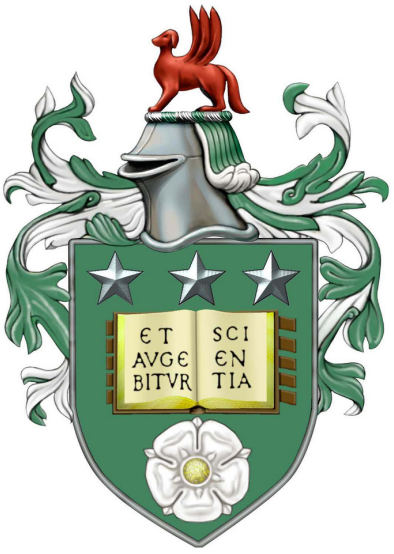
Tracking in the Context of Interaction

by

Aryana Tavanai

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



The University of Leeds
School of Computing
February 2016

Declarations

The candidate confirms that the work submitted is his/her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

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

Aryana Tavanai, Muralikrishna Sridhar, Eris Chinellato, Anthony G. Cohn, and David C. Hogg. Joint Tracking and Event Analysis for Carried Object Detection. *Proceedings of the British Machine Vision Conference (BMVC), pages 79.1-79.11. BMVA Press, September 2015.*

Aryana Tavanai, Muralikrishna Sridhar, Feng Gu, Anthony G. Cohn, and David C. Hogg. Carried Object Detection and Tracking using Geometric Shape Models and Spatio-Temporal Consistency. *Proceedings of the 9th International Conference on Computer Vision Systems (ICVS), volume 7963 of Lecture Notes in Computer Science, 223-233. 2013.*

Aryana Tavanai, Muralikrishna Sridhar, Feng Gu, Anthony G. Cohn, and David C. Hogg. Context Aware Detection and Tracking. *In Proceedings of the 22nd International Conference on Pattern Recognition (ICPR), 2197-2202. 2014.*

The candidate confirms that the above jointly-authored publications are primarily the work of the first author. The role of the other authors were mostly editorial and supervisory.

Abstract

Detection, tracking and event analysis are areas of video analysis which have great importance in robotics applications and automated surveillance. Although they have been greatly studied individually, there has been little work on performing them jointly where they mutually influence and improve each other. In this thesis we present a novel approach for jointly estimating the track of a moving object and recognising the events in which it participates.

The contributions are divided into three main chapters. In the first, we will introduce our geometric carried object detector which allows to detect a generic class of objects. This detector primarily uses geometric shape models instead of using pre-trained object class models and does not solely rely on protrusion regions.

The second main chapter presents our spatial consistency tracker which incorporates events at a detection level within a tracklet building process. This tracker enforces spatial consistency between objects and other pre-tracked entities in the scene.

Finally, in the third main chapter we present our joint tracking and event analysis framework posed as maximisation of a posterior probability defined over event sequences and temporally-disjoint subsets of tracklets. In this framework events are incorporated at a tracking level, where tracking and event analysis mutually influence and improve each other.

We evaluate the aforementioned framework using three datasets. We compare our detector and spatial consistency tracker against a state-of-the-art detector by providing detection and tracking results. We evaluate the tracking performance of our joint tracking and event analysis framework using tracklets from two state of the art trackers, and additionally our own from our spatial consistency tracker; we demonstrate improved tracking performance in each case due to jointly incorporating events within the tracking process, while also subsequently improving event recognition.

Acknowledgements

The Ph.D. journey is like hiking with a group of people where you have to traverse a region with many peaks and valleys. During the more challenging moments, it is the mutual help and encouragement between the members of the group that makes the journey easier and bearable. Given human nature, this aid is naturally given when a person requires it, whether he/she has asked for it or not. As Saadi has beautifully put it:

Human beings are members of a whole,
in creation of one essence and soul.

If one member is afflicted with pain,
other members uneasy will remain.

Saadi (1184-1291)

Although one page is not enough to express my gratitude, here I would like to thank the people that have helped me during my journey and moments of *pain*. First of all to my supervisors Anthony Cohn and David Hogg for giving me the opportunity to take on this journey. Your invaluable feedback and advice in addition to your encouragement of me pursuing novel ideas has made me the researcher I am proud of being today. A special thank you to Muralikrishna Sridhar, a colleague whom I have never enjoyed working with more. I greatly appreciate your guidance and help with me developing novel ideas and the skill of critical thinking. Also a big thank you to Roger Boyle for introducing me to the world of Computer Vision and giving me great insight on the different directions I could take during the early stages of my studies and my career.

I would also like to thank two very important people. First and foremost, Dr. Christiana Panayi who has helped me at every step of this journey. I greatly appreciate your patience and support, specially considering you were going through the same journey. Let it also be recorded in history that her cakes were legendary. Also a big thank you to my dear friend and colleague Dr. Tim Yang, where we shared an even longer journey starting in 2006. We made it Tim ... we made it. Thank you for being a great brother!

Also a big thank you to all ex-Ph.D. students and staff that have had great influence on me, but have now gone to better places, including but not limited to, Sam “CmdrSammo” Johnson, Feng “FengyBoy” Gu, Krishna Sandeep Reddy Dubba, Constantine Zakkaroff, Patrick Ott, Eris Chinellato, Adam Johns, Dimoklis Despotakis, Keeran Brabazon, Ian Hales, Nicole Kerrison and Ardhendu Behera. The same goes to the current cohort, Jawad, Muhannad, Paul, Han, Alicja, Matt, Dave, Duane, Fouzhan, Owais and Yiannis. Thanks for making the Ph.D. a much more memorable and enjoyable time. I would also like to

gratefully acknowledge the financial support of the Mind's Eye project VIGIL (W911NF-10-C-0083), the EU-FP7 projects RACE (287752) and STRANDS (600623) and thank all the members involved.

Most importantly however, I would like to thank my parents and my sister. I will never forget your sacrifices that allowed me to be in the position I am today. My mother who through her untiring efforts helped me during all of my studies and made sure I succeed at every stage of my life. My father who has been my role model ever since I was a child thus motivating me to purse a scientific career in his footsteps. My grandparents whom I dearly love for their never ending support and encouragement. Last but not least, my sister who has added much joy to my life ever since we were children. Where would I be without your words of wisdom?

Dedicated to my sister Tanya

Contents

List of Figures

List of Tables

List of Notations

Latin

A - Lower asymptote in generalised logistic function

B - Growth rate in generalised logistic function

b - An object boundary

$\mathcal{C}$ - Cost of a detection d

C - Typically takes a value of 1 in generalised logistic function

$\mathcal{C}_c$ - Cost for edge chain connectivity

$\mathcal{C}_{dm}$ - Cost for tracklet distance term

$\mathcal{C}_g$ - Cost for geometric shapes

$\mathcal{C}_h$ - Cost for heatmap

$\mathcal{C}_p$ - Cost for protrusion mask

$\mathcal{C}_{pc}$ - Cost for path continuity

$\mathcal{C}_r$ - Cost for the object-entity relationship

$\mathcal{C}_s$ - Cost for tracklet smoothness

$\mathcal{C}_{sc}$ - Cost for tracklet shape continuity

D_1 - Function for calculating the shortest euclidean distance between two lines

D_2 - Function for calculating the Euclidean distance between two points

D_3 - Function for calculating the Euclidean distance between a point to a line

d - A carried object detection

d_{gt} - A ground truth carried object detection

$\mathcal{E}$ - Set of event types

E - Number of event labels

$\boldsymbol{e}$ - Mathematical constant

e - An individual event type

$\mathcal{F}$ - Set of all foreground regions

F - Smoothing function

f - An individual foreground region

g - A co-linear group of edge lines

H - Heatmap

$\mathcal{I}$ - A video sequence

I - An image frame

$\mathbb{I}$ - A difference image

$\mathbb{I}'$ - Gray-scale foreground image

$\mathbb{I}''$ - Binary foreground image

$\mathcal{K}$ - The maximum level in level-wise mining

K - Upper asymptote in generalised logistic function

k - A level in level-wise mining

L - Set of all edge lines

$\mathcal{L}$ - Set of all edge chains

$\mathbb{L}$ - Set of all possible edge chains

l - A subset of edge lines

M - Starting time in generalised logistic function

$\mathcal{N}$ - Normal Distribution

N - Total number of Frames

O - Detection observations in the switched Kalman filter

$\mathcal{P}$ - Power Set

$\mathcal{Q}$ - Function to create HMM observation matrix

Q - Variable defining a parameter in the generalised logistic function

$\mathcal{R}$ - Set of reference entity tracks

r - An individual reference object track

r^{μ} - Average bounding box of a reference track

S - A sequence of event states

$\mathcal{S}$ - All possible event sequences

s - A single Event

$\mathcal{T}$ - Set of all object tracklets

$\mathbb{T}$ - Set of all possible tracklets

T_ω - A candidate object track

$\mathcal{V}_\mathcal{T}$ - Set of all detections not part of a tracklet

v - Variable affecting asymptote maximum growth in generalised logistic function

x - Variable for values taken as input by the generalised logistic function

X - Observation matrix for HMM

Z - Function for calculating the angle between two lines

Greek

α - Width ratio

β - Height ratio

δ - Heatmap offset function

Δx - Relative offset in the x dimension

Δy - Relative offset in the y dimension

η - Span length of the smoothing function

γ - Overlap threshold

ω - A subset of tracklets

μ - Mean

Σ - Covariance matrix

σ - A generalised logistic function

τ - A single tracklet

θ - A set of parameters for the generalised logistic function

θ_a - Angle model for normal distribution

$\theta_{\mathbf{c}}$ - Spatial consistency parameters

$\theta_{\mathbf{con}}$ - Convex shape parameters

θ_d - Distance model for normal distribution

$\theta_{\mathbf{dm}}$ - Distance parameters

$\theta_{\mathbf{el}}$ - Elongated shape parameters

$\Theta_{\mathbf{g}}$ - Geometric shape models

$\theta_{\mathbf{pc}}$ - Path continuity parameters

$\Theta_{\mathbf{s}}$ - Tracklet suitability model

$\theta_{\mathbf{s}}$ - Spatial term parameters

$\Theta_{\mathbf{st}}$ - Set of parameters for spatial and temporal terms

$\theta_{\mathbf{sc}}$ - Shape continuity parameters

$\theta_{\mathbf{t}}$ - Spatial term parameters

λ_k - Smallest level for accepting detections in level-wise mining

ψ - An angle

List of Acronyms

CRF - Conditional Random Field

DHD - Damen and Hogg Detector

GMM - Gaussian Mixture Model

HMM - Hidden Markov Model

JTEA - Joint Tracking and Event Analysis

MAP - Maximum a Posteriori

MCMC - Markov Chain Monte Carlo

MCMCDA - Markov Chain Monte Carlo Data Association

MIL - Multiple Instance Learning

MRF - Markov Random Field

PCA - Principal Component Analysis

PD - Periodicity Dependency

SCT - Spatial Consistency Tracker

Chapter 1

Introduction

When a person is asked to observe and describe a scene, no matter how complex the environment, they can easily transform the visual information into a sequence of meaningful abstractions. In nearly all cases, these abstract definitions of the scene can be described in terms of *interactions* between one or more entities and objects. Whether it is two people talking to each other, a car over taking another car, a person carrying their briefcase to work or even a single person waiting, these entities and objects are, or will interact with others at a certain point in time.

The field of scene understanding in computer vision aims to tackle this challenge for an observing machine, allowing it to understand and describe such interactions from a video. The Oxford dictionary defines the word interaction as a "reciprocal action or influence" [67]. That is, the interacting parties are bound to each other and are equally involved in the interaction, where they also mutually influence each other. Therefore, for a machine to understand and describe an interaction, it must answer three questions (i) *who* are the parties involved? (ii) *what* is the nature of the interaction? and (iii) *how* are they influencing each other?

The area of *Tracking* in computer vision aims to solve the first question, i.e. the *who*. Here, the goal is to identify the objects and entities that are interacting with each other. This is usually done by detecting or segmenting the objects and entities in the scene and tracking them throughout the video.

The areas of *Activity Recognition* and *Event Analysis* aim at solving the second question, i.e. the nature of the interaction. These areas attempt to identify the interaction that is

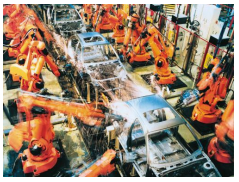
	Entity-Entity Interaction	Object-Entity Interaction	Object-Object Interaction
One to One	 Handshake	 Pickup	 Overtake
Many to One	 Carry	 Hold	 Pull
Many to Many	 Dance	 Snowball Fight	 Build

Figure 1.1: Examples of the three types of interaction between entities and objects, namely entity-entity, object-entity and object-object interactions. Each row highlights examples of the aforementioned interactions based on the relationship between the entities and objects, namely one-to-one, many-to-one and many-to-many.

occurring between the interacting objects and entities. This typically involves labelling parts of the video in terms of the occurring interactions (also referred to as *activities* or *events*), e.g. talking, over taking, carrying or waiting based on the interaction examples previously provided. In this work, as illustrated in Figure 1.1, we categorise these interactions into three main groups namely *entity-entity*, *object-entity* and *object-object* interactions.

In the examples provided in Figure 1.1, an entity is primarily, but not limited to, a person performing an interaction while an object can be anything that is interacted with. However, in the object-object case, an object may take the role of interacting with another object, e.g. robots interacting with a motionless child in the *Pull* example. Based on this example and other interactions not involving an interacting person, since the term *activity* is primarily focused on a participating person, to capture all types of interactions we will henceforth use the general term *events*, which we will use interchangeably with the term interaction. Additionally in Figure 1.1, the entities or objects participating in such events

may have a *one-to-one*, *many-to-one* or a *many-to-many* relationship. It is also worth noting that a many-to-many relation may be composed of shorter atomic one-to-one relations.

The last question, how two parties influence each other to perform an interaction, is answered by analysing both the tracking solution of question one and the event analysis solution of question two. Areas of *Machine Learning* and *Knowledge Representation* aim at solving this problem by taking into account the distinct behaviour of the tracking solution of the two parties with respect to each other. Their mutual influence on one another determines the interaction they undergo, which is defined by a label as part of the solution to the second question.

In this thesis we explore and investigate the benefits of incorporating the knowledge of interactions within tracking. We tackle each of the aforementioned three questions and provide a single framework that simultaneously solves both the tracking and the event analysis problem. Most importantly however, this framework incorporates the learnt distinct behaviour of the tracking solution and the characteristic properties of each interaction in order to mutually influence and improve both the tracking and the event analysis solutions. In the next section we present our main goal.

1.1 Goal

The main goal of the work presented in this thesis is *tracking in the context of interaction*. This goal primarily focuses on incorporating the knowledge of events into object tracking. More specifically, we use a tracking solution to obtain an event solution, and then use the event solution to improve the tracking solution. We approach this goal based on the following methodology:

1. We capture and model the nature of each event based on the notions of consistency and inconsistency.

2. We then use these learnt event models to perform event analysis and exploit the knowledge of events in order to improve the outcome of tracking.

3. We then use the improved outcome, i.e. the tracks, to improve event analysis.

4. Finally event analysis would be applied on tracks they aim to improve and tracking would be applied by taking into account the events they have produced.

In the next section we provide a brief summary of our approach in accomplishing and solving the above goals.

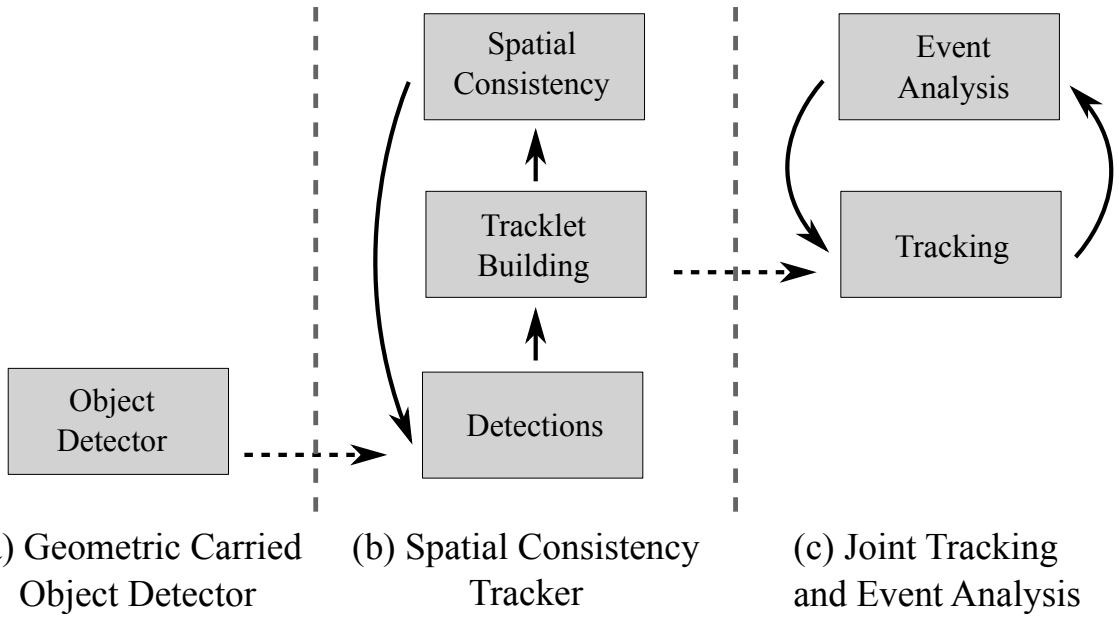


Figure 1.2: The three main frameworks of this thesis namely (a) Geometric Carried Object Detector (b) Spatial Consistency Tracker and (c) Joint Tracking and Event Analysis. Dotted arrows represent the link between the output of one framework to the input of another. The solid arrows represent internal links between elements within a framework and the notion of influence between them.

1.2 Approach

In this thesis, our approach is primarily focused on incorporating interactions of the object-entity type within the tracking solution. However, our work is not limited to such interactions and can in theory be applied to other aforementioned types presented in Figure 1.1. We also particularly focus on event-based tracking for domains which tend to contain generic objects for which it is not straightforward to train class specific object detectors. Therefore, the task of object tracking becomes especially challenging in domains such as *carried objects*, as false, partial and missing detections are highly prevalent [24, 28]. This leads to false tracks, and also tracks that are heavily fragmented. We observe this phenomenon when we apply state-of-the-art trackers by Andriyenko et al. [4] and Pirsiavash et al. [63].

Therefore, to overcome the goals described in Section 1.1 in such domains, we apply the following three frameworks, illustrated in Figure 1.2, each of which focuses on a particular part of the *tracking in the context of interaction* goal:

(a) **Geometric Carried Object Detector:** As a requirement for tracking generic objects which are hard to model, we require an object detector to initially detect such

objects. We therefore propose our own object detector that uses simple geometric shape models to provide detections for a large range of carried object types. These detections are then used as input in the next framework.

(b) **Spatial Consistency Tracker:** To minimise the effects of false positives and maximise the affects of true positives detections within the final third framework, the role of the second framework is to locally connect detections and form tracklets. More importantly however, this is done while taking into account the spatially consistent behaviour between the object and entity performing the interactions to improve the quality of tracklets. These tracklets are then used as input in the next framework.

(c) **Joint Tracking and Event Analysis:** The final framework incorporates both the tracking and the event analysis aspects, where they both mutually influence and improve each other. In this framework events are modelled in terms of both consistency and inconsistency and are directly used to improve the quality of object tracks, satisfying the goals set in Section 1.1.

It is worth noting that any of the three aforementioned frameworks are not limited to each other, and may take detections or tracklets as input from any other state-of-the-art approaches. In the next section we describe the challenges that the above frameworks tackle and overcome.

1.3 Challenges

There are various challenges that require to be overcome to accomplish the outlined goal in this thesis. In this section we present the challenges faced in this work from the point of view of (i) object detection and (ii) joint tracking and event analysis:

(i) Challenges in object detection:
In order to localise and segment an object in an image frame, one may face various challenges. Focusing on carried object detection, these challenges include, but are not limited to:

- **Object Variety:** Objects used in everyday interactions can vary significantly in type whether it is with respect to their size, shape, colour or any patterns they may have. This makes it particularly challenging to learn models for each object type.

- **Occlusion:** Due to the nature of interactions, objects and entities are in close proximity, greatly increasing the chance of occlusion. This makes detectors unable to detect objects for intervals of varying length, depending on the interaction.
- **Lighting Conditions:** Changes in the weather or automatic brightness adjustments on cameras are examples of slow or sudden changes in lighting respectively. This change can heavily effect the performance of object detectors that rely heavily on colour and edges.
- **Clothing:** Items of clothing introduce various complexities into object detection whether it is due to the wide range of patterns on clothing or the creases that naturally emerge on clothes during movement.
- **Anatomical Differences:** If one would like to attempt to detect objects by initially finding the person interacting with it, this may not be straight forward due to physical differences between people.
- **Camera Angle:** Whether it is a top down, bird's-eye or human height level viewpoint, the camera angle in which data is recorded can significantly effect the performance of an object detector.
- **Motion Blur:** Sudden movement of objects may cause motion blur, depending on camera settings, making it harder to find features for object detection.
- **Scene Depth:** In the real world objects continue to have the same size when moved around. However, in the image plane, when an object moves further into the scene it changes size due to scene depth and is harder to detect if the detector does not take this into account.

(ii) Challenges in joint tracking and event analysis:

While traditionally track-based event analysis has been performed after the tracking process has been completed, incorporating event analysis within tracking introduces various challenges. Firstly, event based tracking is a circular problem. This problem involves inferring events using reasonable tracks, and then using these events to subsequently improve the tracks. Due to this challenge, event based tracking has been rarely approached [93, 5, 17]. This becomes even more challenging due to the prevalence of false and fragmented tracks in domains containing generic objects, as a result of using a large number of false positives detections in the tracking process.

Secondly, in attempting to solve the circular problem, one must take into account that there are challenging cases where there might not be a suitable event analysis solution

which the tracker can exploit. This could happen either in the early stages of the tracking process where a suitable tracking solution has not been found, leading to an unsuitable event analysis solution, or even if a suitable tracking solution is found, the event analysis aspect is unable to find a suitable solution. Such cases, as a result of jointly combining tracking and event analysis, must be considered and taken into account.

Finally, and most importantly, to jointly perform tracking and event analysis one must overcome the challenge of finding a suitable way for the tracker and the event analysis technique to *communicate* through in which they can mutually influence and improve one another.

1.4 Thesis Overview

Current state-of-the-art approaches for detecting and tracking a generic class of objects, such as carried objects, primarily employ methods that heavily depend on using protrusion to detect such objects [37, 24], or perform object detection without necessarily localising them [57, 82]. In this thesis, while we propose our own carried object detector that overcomes the limitations of other works and the challenges previously described, we take an additional step and approach object tracking differently, that is, to incorporate event analysis within the object tracking process. The novelty and significance of the work presented in this thesis is described below.

1.4.1 Novelty and Significance

This thesis introduces a novel approach in combining object detection, tracking and event analysis. The following include the novel and significant contributions of this work.

- A novel carried object detector that localises carried objects and is not heavily dependant on protrusion or modelling the entity interacting with it.
- Our detector uses generic shape models and does not require training for specific object models and may be used to detect a large variety of carried object types.
- We present a novel framework for building object tracklets which suppresses false positives while promoting and incorporating the true positive detections deemed weak by detectors within the tracking process. This is done by exploiting and capturing the spatially consistent behaviour between the interacting object and entity.

- We jointly perform object tracking and event analysis within a single novel framework. In this approach tracks and events mutually influence and improve each other.
- We model events in terms of spatial consistency and inconsistency and provide a solution to the circular nature of the problem while using a single objective function where tracking and event analysis can communicate.

1.4.2 Outline

The rest of the thesis is organised as follows:

Chapter 2: Related work
In this chapter we provide a full literature review on any work related to carried object detection and tracking. We also present related work within the areas of tracking and contextual tracking. Since these areas are very large, we primarily focus on only describing state-of-the-art approaches.

Chapter 3: Geometric Carried Object Detector
While there have been various approaches to carried object detection, in this chapter we present our novel approach to detecting carried objects. We provide a step-by-step description of our detector, starting with the use of edges and finishing with object boundaries.

Chapter 4: Spatial Consistency Tracker
As the first step to incorporate the notion of interaction within object tracking, in this chapter we describe our Spatial Consistency Tracker (SCT). This tracker uses the object-entity interaction and models their behaviour via a spatial consistency map. The influence of this map on the tracklet building process of SCT aids in suppressing false positives while promoting weak true positive detections within the tracking process.

Chapter 5: Joint Tracking and Event Analysis
To accomplish the main goal of combining and jointly performing tracking and event analysis, this chapter describes the novel framework that achieves this goal. It includes details on how events were modelled and incorporated within the tracking process. The benefits of our Joint Tracking and Event Analysis (JTEA) framework are highlighted in this chapter by describing the optimisation process that applies this framework.

Chapter 6: Evaluation
We evaluate our carried object detector, our SCT framework and out JTEA framework in this chapter by presenting quantitative and qualitative results. The evaluations are performed on various datasets, one of which was created and made publicly available. We compare our framework against various state-of-the-art approaches to evaluate for both detection and tracking.

Chapter 7: Conclusion and Future Work
The final chapter provides a summary of the work presented in this thesis and a final conclusion on their novelty and significance. Moreover, we also provide potential future extensions and research directions in order to expand on the frameworks presented.

Chapter 2

Related Work

Video analysis, as part of scene understanding, is one of the oldest and most widely applied fields within computer vision. The higher computation power of machines and the availability of inexpensive and high-quality cameras has resulted in a rapid increase of applications within this field. As a result, the process of being able to automatically understand information from videos has significantly improved over the past couple of decades.

While the influence of the above technological advancements has greatly effected the rapid growth of video analysis, the demand for this field has also been a major contributing factor. The need for a robot to understand what it sees due to advancements in robotics, the demand for more accurate and robust security measures in the area of automated surveillance or biometric recognition against any possible threats, the need for automatic annotation and retrieval of videos in the area of scene understanding due to the rapidly expanding multimedia databases and many more, are all reasons why video analysis has been of great importance and has been given much attention in recent years.

According to a survey paper by Yilmaz et al. [101], video analysis consists of three key steps, (i) the detection of interesting moving objects, (ii) the tracking of such objects from frame to frame and (iii) the analysis of object tracks to recognise their behaviour. The contributions of this thesis which are presented in Chapters 3, 4 and 5, employ techniques from each of the aforementioned key steps. As a result, the related literature for each of the key steps have been presented in this chapter.

Section 2.1 presents the related work of the first step of video analysis, object detection.

As described in Chapter 1, rather than presenting related literature on object detection, we only focus and provide a full literature review on carried object detection. Therefore Section 2.1 highlights the successful trends that have emerged in the young but rapidly growing literature on carried object detection and the challenges within this domain.

For the second step, tracking, Section 2.2 initially provides a general overview of various types tracking while finally focusing on a few approaches that are most relevant to our approach.

As part of the third step, analysis of object tracks, our work aims at incorporating context within tracking. However, we primarily focus on event context in order to improve the tracking of the carried objects. Rather than providing a literature review of work in the field of event analysis, in Section 2.3.2 we describe the few existing approaches that incorporate event analysis within their tracking process. We also present other types of context employed to improve detection and tracking in Section 2.3.1.

Finally, based on the aim of this thesis to jointly perform tracking and event analysis, in section 2.3 we also describe related work which improve the process of tracking by using events or any other type of context.

2.1 Carried Object Detection

Carried object detection can be divided into three main types of approaches, namely (i) protrusion based, (ii) model based and (iii) classification based. In the following sections, related work on each of the aforementioned types is provided. At the end of each section a discussion of the benefits and limitations of the approaches is given.

2.1.1 Protrusion Based

The earliest approach to carried object detection aims at initially identifying the person and background regions and then attempting to explain the remaining regions in terms of carried objects. These remaining regions are referred to as *protrusion* regions which are regarded as the part of foreground that is different from the person region.

This approach was first incorporated in *Backpack* by Haritaoglu et al. [37]. Illustrated in Figure 2.1, the authors place a global shape constraint requiring the human body shape to be symmetric around the body axis. This leads to a symmetric human model which will be used to obtain outlier regions based on non-symmetric regions of the person silhouette and the model. Using periodic motion detection [22], outliers from arms and legs tend to be periodic, while continuous outliers from sufficiently large carried objects can be detected

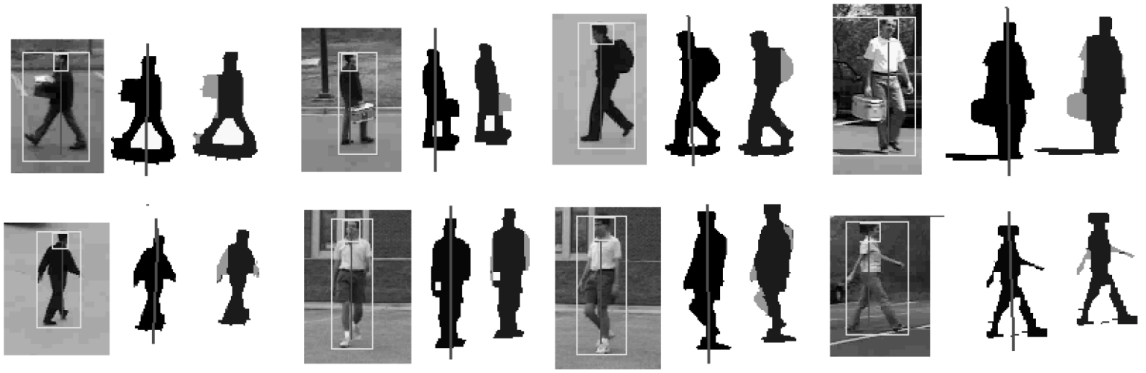


Figure 2.1: Backpack [37]: Examples of symmetry based segmentation of potential carried objects with initial detected head location, computed symmetry axis and final non-symmetric region segmentation. Combined with periodicity analysis these regions may be considered as carried object regions.

due to continued symmetry constraint violations. These continuous outlier regions are segmented and defined as a carried object.

A similar approach to the above is taken by BenAbdelkader and Davis [6], additionally however, they use the gait of a person to incorporate spatio-temporal constraints that are satisfied if a person is naturally walking, but not if a person is carrying an object. They show that since the gait of the person can vary significantly whether the person is walking or carrying an object [44], this knowledge can be used for carried object detection.

Lee and Elgammal [49] also use a protrusion based approach while incorporating a pose preserving dynamic shape model technique. This technique supports pose-preserving shape reconstruction for various people, views and body poses. An iterative estimation of this model allows for a better estimation of outliers (protrusion) in addition to accurate body pose.

The most recent extension of protrusion based approaches introduces refinements such as 3-D exemplar temporal templates corresponding to different viewpoints of a walking person together with spatial priors in recent work by Damen and Hogg [23, 24]. Illustrated in Figure 2.2, the foreground blobs obtained from background subtraction are centred and aligned providing a temporal template. An exemplar temporal template is then transformed (translation, scaling and rotation) to best match the obtained temporal template. By comparing the temporal and exemplar template, protruding regions are found. A Markov random field with a trained spatial prior is then used to segment carried objects.

A few approaches have improved on the Damen and Hogg approach. Yuan et al. [103] use Principal Component Analysis (PCA) and exhaustive search for temporal template matching and perform a fuzzy clustering method to classify protruding pixels. Tzanidou

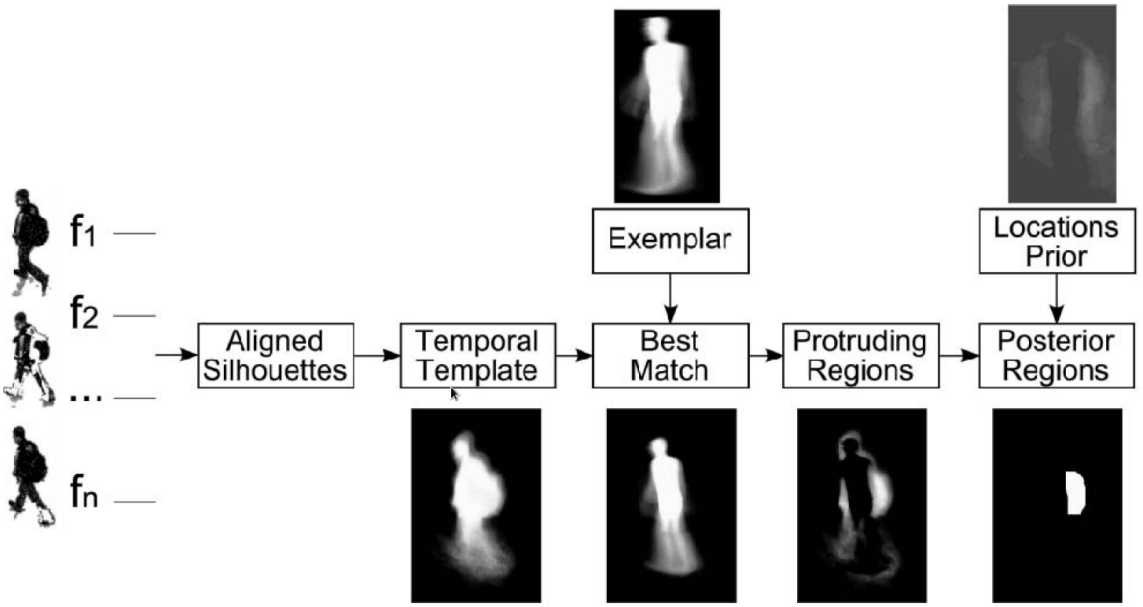


Figure 2.2: The pipeline of segmenting the carried object region in Damen and Hogg [24]. By comparing the temporal template of a person with a best matched exemplar template, protrusion regions are found which lead to carried object detections.

et al. [89, 90] use colour information and movement direction to improve detection and additionally perform baggage type classification.

Discussion

While the above protrusion based approaches reasonably find protruding carried objects, they are unable to find the object if it is on the person region. The approaches also rely heavily on fitting person models to silhouettes and may additionally require camera parameters.

2.1.2 Model Based

Unlike previous protrusion based approaches, a supervised approach is adopted by Branca et al. [12], demonstrating that pre-trained object-class models for specific types of objects may be useful in domains where the variety of carried objects is relatively small, known in advance, are of sufficient size and there is limited clutter in the background. In their work, they use patterns on the person region, represented via coefficients of their wavelet decomposition, and classify these patterns using a supervised three layer neural network.

A different approach is taken by Chuang et al. [18] where they detect a carried object by comparing histograms of before and after a person possessing an object. In this approach

(a) (b) (c)

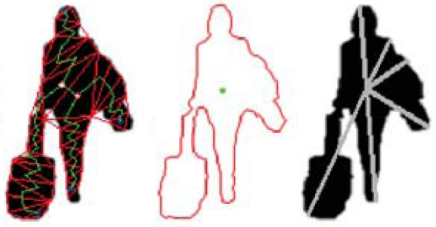
Figure 2.3: Carried object detection using ratio histograms [18]. Colour profile of the carried object is obtained by comparing the foreground of the person with and without the object.

they obtain the colour profile of the carried object and then use a Gaussian mixture model to segment the object from its background. Sample images illustrating this approach are presented in Figure 2.3.

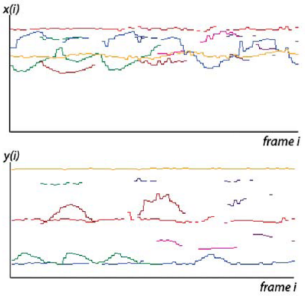
Chayanurak et al. [16] use a star skeleton approach for carried object detection illustrated in Figure 2.4. To obtain the star skeleton, Figure 2.4a from left to right, they initially obtain a Delaunay triangle mesh of the human shape, similar to [19], and extract a triangle-based skeleton which provides the centroid of the person. They then obtain a distance from the centroid to each human contour point. After smoothing the distances, they obtain local maxima or peak points. Each of these peak points (or limbs) is then connected to the centroid, creating the star skeleton.

Each limb is then tracked throughout the video sequence in terms of its x and y positions against time resulting in graphs shown in Figure 2.4b. Based on this graph, any tracked limb with a motion less than a certain threshold is classified as a carried object. In order to determine the boundary of the carried object, the silhouettes feature information based on sink curves adjacent to the tracked points are used. Examples of these adjacent sink curves are illustrated in Figure 2.4c, which define the bounding box of the carried objects displayed in Figure 2.4d.

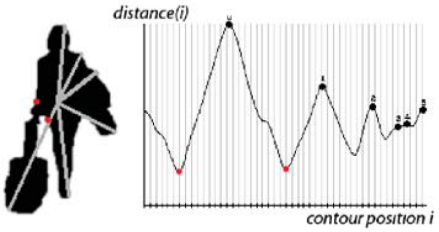
Although this approach is not based on protrusion, it can only obtain carried object detections that are outside of the person region. Moreover, this approach is not suitable in cases where the carried object is moving relative to the person (e.g. swinging or putting



(a) Obtaining star skeleton.



(b) Tracking the limbs of the star skeleton.



(c) Finding the sink curves of the detecting carried object limb.

(d) Boundary of the carried object limb based on the sink curves.

Figure 2.4: Chayanurak et al. [16] carried object detection using star skeleton.

down) as the object limb will have high motion.

A recent multi-model based carried object detector is developed by Dondera et al. [28]. Illustrated in Figure 2.5, they use three types of detectors, namely (i) optical flow-based protrusion, (ii) segmentation-based colour contrast and (iii) occlusion boundary-based moving blob detectors, and combine them under a minimally supervised framework. Their approach to carried object detection is to disambiguate between the obtained regions (from the aforementioned detectors) corresponding to body parts/noise versus those that are carried objects, based on the context of the human silhouette.

The optical flow-based protrusion detector (Figure 2.5a) builds a *carried probability mask* that reflects how close the motion of a pixel is to the average motion within a human bounding box. This gives rise to potential protruding carried objects.

The segmentation-based colour contrast detector (Figure 2.5b) uses mean shift clustering on the foreground mask. This provides numerous segmentations of potential objects of which their colours stand out against the human silhouette.

As occlusion is highly prevalent in carried object detection, the last detector, occlusion boundary-based moving blob detector (Figure 2.5c), aims at detecting occlusion boundaries

(a) Optical flow-based protrusion detector

(b) Segmentation-based colour contrast detector

(c) Occlusion boundary-based moving blob detector

Figure 2.5: Dondera et al. [28] carried object detector with minimal supervision. In this work the authors combine three types of detectors under a MIL framework

that cover an occluded object. To obtain these regions they apply the work of Sundaram et al. [80] and obtain occlusion boundaries defined as a group of pixels where the flow forward of a frame is inconsistent with the flow back into the frame, or where the flow gradient has a large magnitude.

Each of the regions obtained from the above detectors is then given to a Support Vector Machine (SVM) classifier to filter out non carried object regions. This classifier uses two types of features characterizing (i) the shape of a region and (ii) the relation of the region to the human silhouette. This is accomplished within a Multiple Instance Learning (MIL) framework to learn a model for carried object regions where human track intervals are labelled as *carry* (carried object is present) and *walk* (carried object is not present).

Discussion

Object detection based on trained models are effective when one knows what object types to expect. However, in surveillance or long term applications within robotics, it is difficult to predict what kind of objects will be present in the scene. Therefore the aforementioned approaches are limited to detecting only the objects they train for and expect beforehand. Moreover, in some approaches the modelling heavily depends on the person region, colour and protrusion.

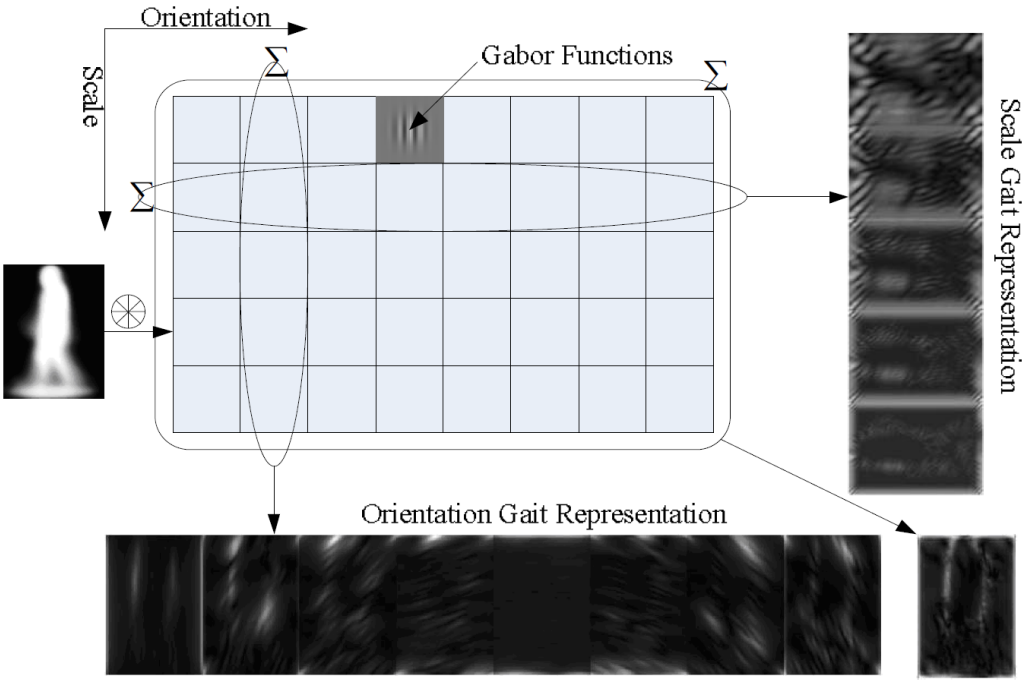


Figure 2.6: Three types of Gabor representations used by Tao et al. [82], namely orientation, scale and total for classification of the carrying object status.

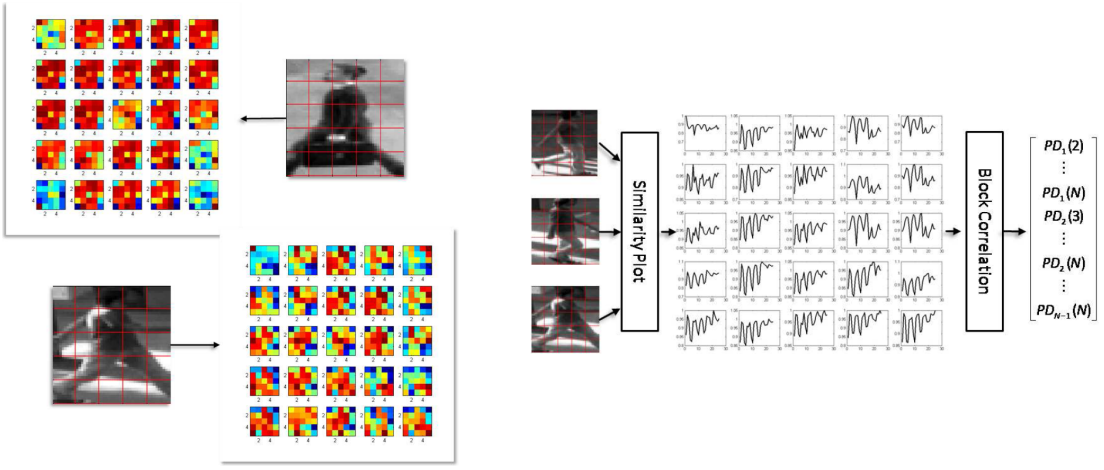
2.1.3 Classification Based

Indirect approaches for carried object detection are also employed where a pre-trained appearance model of *person without carried objects* have been built and *person carrying objects* are detected as anomalies, but without localising the object. One of the earliest of these approaches is by Nanda et al. [57] where they use a two layer neural network for binary object classification of (i) pedestrian or (ii) pedestrian with shape outliers.

Tao et al. [82] take a similar approach, instead however, they use a Gabor gait based representation as their features. Rather than using standard representations of gait images, as illustrated in Figure 2.6, they introduce and use three types of Gabor representations, namely orientation, scale and total gait representation. They then apply a general tensor discriminant analysis for classification to solve the *carrying status* problem.

Qi et al. [64] also perform classification to detect whether a person is carrying an object or not. However, unlike previous classification approaches they localise the carried object by finding the out-most point in the human contour.

Various alternative classification approaches are adopted by Senst et al. [71, 72, 73] which heavily focus and take advantage of the motion of the person. In [71] they define a periodicity dependency (PD) descriptor that describes the spatial dependency of human



(a) (b)

Figure 2.7: The use of periodicity dependency by Senst et al. [71] for carried object detection.

motion, such as synchronous arm and leg motion. Illustrated in Figure 2.7, the movement of different body parts are highly dependant on each other due to the kinematic chain formed by the human body. By dividing the bounding box of the person into various blocks, a spatial map of self-similarities between the blocks using the PD descriptor can be obtained. Each block may have one of three signal types; (i) blocks containing body parts exhibiting a cyclic motion have a periodic signal, (ii) blocks with static body parts have a quasi-linear signal and (iii) blocks containing the carried objects have a cyclic motion, however with minor amplitude compared to blocks with body extremities. They then classify each frame based on the descriptors by providing a binary class, and then use a voting system to classify the entire sequence, whether a person is carrying an object or not.

In [72], Senst et al. use motion statistics based on optical flow to classify whether a person is carrying an object. They use a Gaussian mixture motion model (GMMM) and define descriptors based on speed and direction, independent of motion, to detect carried objects as regions not fitting in the motion description model of an average walking person. In [73] they take a similar optical flow based approach while incorporating Lagrangian Dynamics for the purpose of modelling the appearance of pedestrians.

Discussion

Classification approaches provide a suitable alternative approach to detecting carried objects. However, the majority of these approaches cannot localise the object which makes the task of tracking and event analysis more challenging.

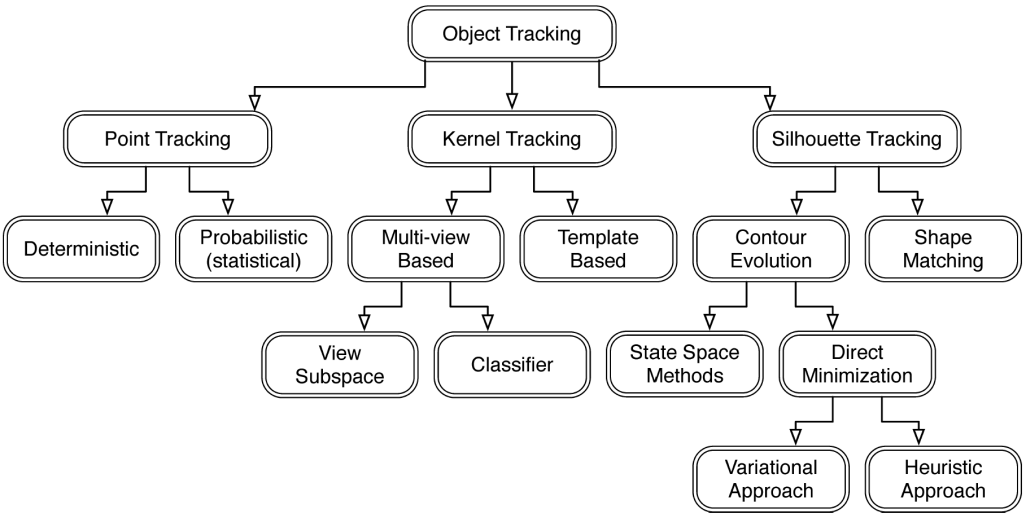


Figure 2.8: Taxonomy of tracking methods [101].

2.2 Tracking

The goal of the object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. Described by Yilmaz et al. [101], tracking requires two tasks, (i) detecting the object and (ii) establishing correspondence between the object instances across frames. These two tasks can be performed either separately or jointly. If done separately, object detection is initially completed and then the tracker corresponds the obtained object detections across frames. If done jointly, the object locations and the correspondence is jointly estimated by iteratively updating object location and region information from previous frames.

Figure 2.8 presents a taxonomy of tracking methods [101]. Here object tracking is divided into three main categories namely *point tracking*, *kernel tracking* and *silhouette tracking*. Each of these categories contain a large body of work where it would not be possible to cover all in this thesis. Since we use our own external detector to provide detections, in this thesis we pursue a tracking by detection approach which follows similarly to point tracking. Our approach is in agreement with the majority of recent tracking approaches [60, 33, 96, 51, 2, 13]. Trackers that simultaneously perform detection are not always able to handle re-initialisation when a target has been lost and may additionally face excessive drift [21, 4]. Therefore similar to our approach, we primarily focus on point tracking where we present a few of the relevant work within this category.

In *point tracking*, object detections in consecutive frames are represented by points and the tracking problem is formulated as the correspondence between them. Although there

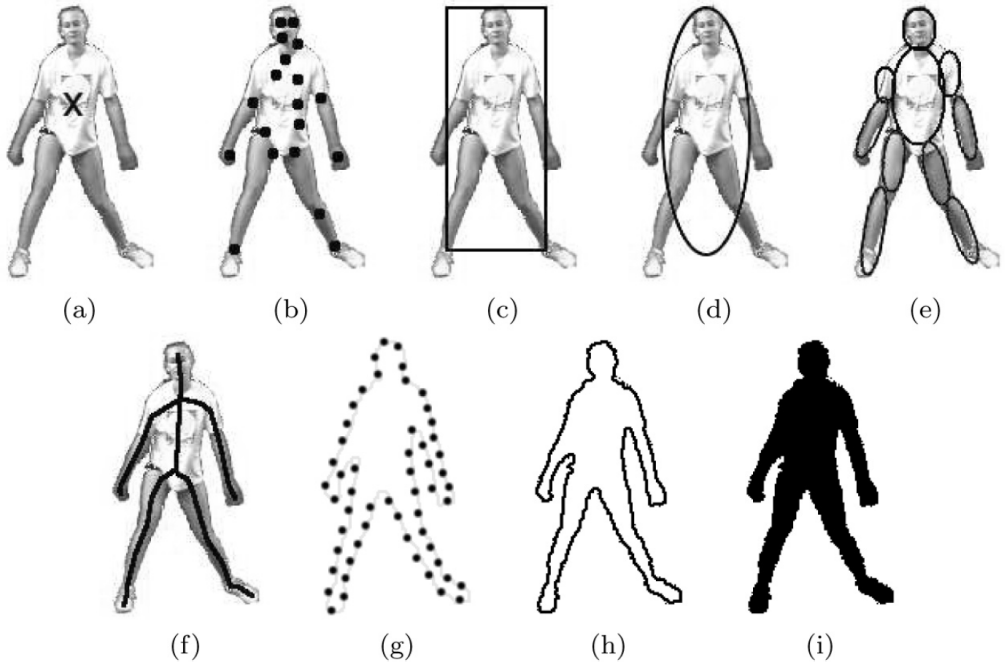


Figure 2.9: Object representations categorised by Yilmaz et al. [101]. (a) Centroid, (b) multiple points, (c) rectangular patch, (d) elliptical patch, (e) part-based multiple patches, (f) object skeleton, (g) complete object contour, (h) control points on object contour and (i) object silhouette.

are various other ways in representing objects, as illustrated in Figure 2.9, the points we refer to define the centre of the bounding box representing the object detection. Tracking through point correspondence is a complex and challenging problem due to the presence of occlusions, missed detections, entries and exits of objects. Point correspondence methods are divided into two categories, namely deterministic and statistical methods. While deterministic [70, 92] and statistical methods [14, 74, 79] have been in use since the 1980s, we focus on more recent related work.

One of the most widely used tracking algorithms is the globally-optimal greedy algorithm developed by Pirsiavash et al. [63]. This work, following the min-cost flow algorithm of [104], formulates the problem of tracking multiple objects in terms of using a cost function that requires estimating the number of object tracks in the scene, as well as their birth and death states. In this approach, they use a greedy successive shortest-path algorithm where the optimal interpretation of a video with $k + 1$ tracks can be derived by locally modifying the solution of k tracks. This process is illustrated in Figure 2.10 where an initial 3-track estimate is present on the left image. With the knowledge that an additional object is also present, they modify the 3-track estimate using a shortest-path/min-flow computation

that pushes the flow from a source to a terminal, as illustrated in the middle image. The solution can then reverse flow along existing tracks to change their segments, producing the 4-track estimate in the right hand side image. Therefore, to find a globally optimal solution, using the network model illustrated in Figure 2.11, a greedy algorithm based on dynamic programming is applied that sequentially instantiates tracks using shortest path computations on the flow network.

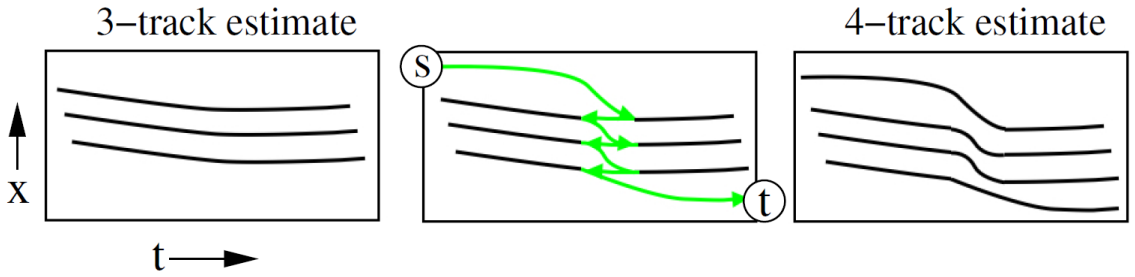


Figure 2.10: Tracking process of Pirsiavash et al. [63] where they derive an optimal interpretation of a video with $k+1$ tracks by locally modifying the solution of k tracks. In this example they use a 3-track estimate and obtain a 4-track estimate by using a shortest-path computation that pushes the flow from a source (s) to a terminal (t).

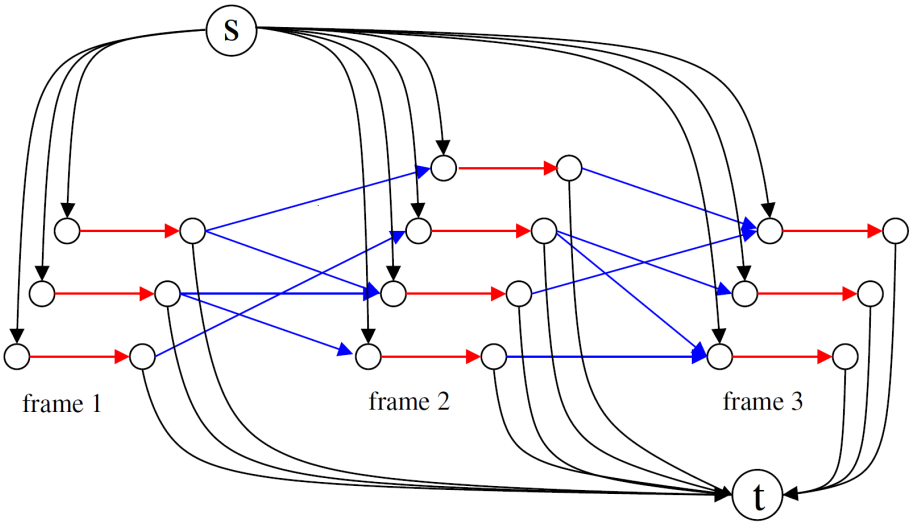


Figure 2.11: The network model of Zhang et al. [104] used by Pirsiavash et al. [63] to find a globally optimal solution. This is done by applying a greedy algorithm based on dynamic programming that sequentially instantiates tracks using shortest path computations on the flow network. Each space-time location is represented by a pair of nodes connected by a red edge. Possible transitions between locations are represented by blue edges. To enable tracks to start and end at any spatio-temporal point in the video, each node is connected to both a start node (s) and a terminal node (t).

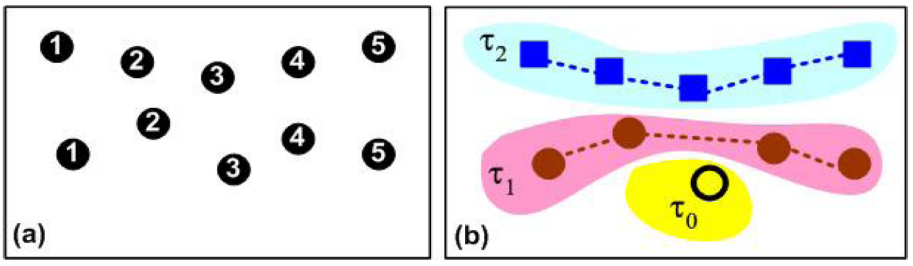


Figure 2.12: The tracking process of partitioning observations in image (a) into tracks in image (b) [59]. The number in each observation node represent its frame number.

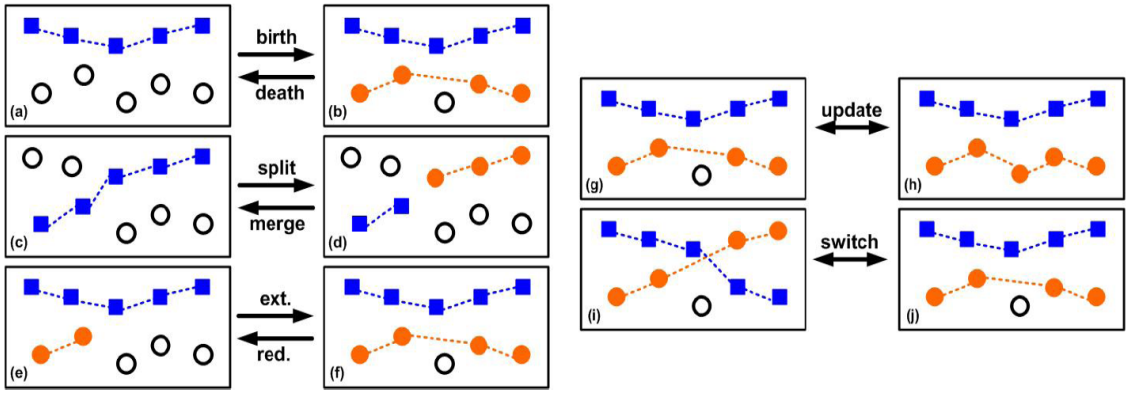


Figure 2.13: The set of moves applied in the Markov chain Monte Carlo data association tracker by Oh et al. [59] to create or modify track partitions of observations.

Similar to point tracking one may consider the point correspondence problem as a data association problem. The Markov chain Monte Carlo data association (MCMCDA) algorithm by Oh et al. [59] produces a tracking solution by partitioning observations into tracks, as illustrated in Figure 2.12. MCMCDA accomplishes this partitioning by approximating the optimal Bayesian filter using a Markov chain Monte Carlo (MCMC) sampling instead of the traditional Bayesian approaches where the optimal filtering prediction is found by summing over all possible associations, weighted by their probabilities [20, 68].

To create or modify new track partitions in the MCMCDA approach various MCMC *moves* are applied. These moves, illustrated in Figure 2.13, are chosen randomly based on a distribution. In a simulated annealing approach, a new track partitioning is obtained as a result of applying a certain move where it may be accepted depending on the *maximum a posteriori* (MAP) estimate and the MCMC acceptance probability. While MCMC has a long history of being used in tracking [62, 8] other notable approaches incorporating MCMC to solve the data association problem include Khan et al. [43] which use MCMC within a particle filter and Yu et al. [102] which take spatio-temporal information into

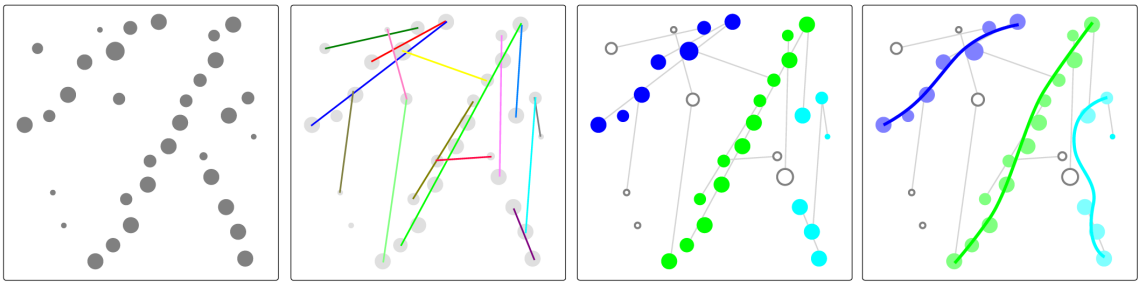


Figure 2.14: The discrete-continuous tracking process of Andriyenko et al. [4]. Given a set of unlabelled object detections in (a) and a set of possible trajectory hypotheses in (b) this tracking process assigns labels for all detections, presented in (c), and re-estimates the trajectories, presented in (d). This process alternates between discrete and continuous tracking where they are performed jointly using an energy minimisation approach.

account.

Another approach to tracking is the discrete-continuous optimisation approach by Andriyenko et al. [4]. This work aims at tackling both the discrete and continuous challenges within tracking. The discrete case addresses the data association problem of tracking where each detection is labelled as belonging to a certain track or being a false positive. In the discrete case however the tracker is limited to a *discrete space* of detections and may be limited in accuracy. The continuous case approaches tracking as finding object trajectories in a *continuous space*, where a trajectory is not necessarily limited to detection locations and is able to more accurately represents the object with respect to its motion and velocity.

While there has been various work in literature tackling the discrete [3, 7] and continuous [50, 97] problems individually, very few have approached combining both approaches. While the aforementioned MCMC approaches [43, 58, 59] to some extent bridge the gap between discrete and continuous aspects, they are limited in terms of the expressiveness of their underlying model. Therefore the state-of-the-art approach by Andriyenko et al. [4] formulates data association (discrete aspect) and trajectory estimation (continuous aspect) jointly as a minimisation of a consistent discrete-continuous energy, building upon the energy minimisation approach of Delong et al. [26]. The tracking process alternates between discrete and continuous tracking.

As illustrated in Figure 2.14, given a set of unlabelled object detections (a) and a set of possible trajectory hypotheses in (b), the tracking process of Andriyenko et al. assigns labels for all detections (c) and re-estimates the trajectories, presented in (d). The label assignment of detections is performed using an energy minimisation approach

considering labelling costs. The trajectory fitting aspect is performed by fitting B-splines to the labelling assignment through an energy minimisation which takes into account how well the trajectory fits the assignments.

Discussion

The aforementioned trackers are very suitable techniques for tracking objects that are to some extent reliably detected. However, when we apply them within the domain of carried objects which contain a large number of partial detections and false positives with a variety of detection strengths, they produce a large number of false positive short tracks (also referred to as tracklets). This is due to the trackers not being built for the purpose of carried object tracking. However, there are other approaches to tracking which take context into account to improve the tracking process in such or similar domains. In the next section we provide a few of these approaches.

2.3 Context Based Detection & Tracking

In this section we present related work which have incorporated context within the tracking process. We divide contextual information into two groups, namely scene and event context, each described in the following sections.

2.3.1 Scene Context

Some of the earliest work on incorporating context within detection include the use of graphical models such as Markov random fields (MRF) and conditional random fields (CRF). In these approaches, during detection, information about pixels surrounding a scanning-window detection is taken into account and thus incorporating contextual information. This type of approach is seen in the work of Torralba [86] and Wolf et al. [95] where they build a representation of context from low level features and use them to facilitate object detection.

Similarly Shotton et al. [75] propose *textons* features that model shape, texture and context in a CRF to segment an image into semantic categories by exploiting context. An example of this work is illustrated in Figure 2.15, where for an image (a) they obtain a texton map (b). Then, based on the feature pair of a white rectangle r and a *grass* texton patch t (c), they obtain feature responses based on locations i. In this example i_1 (d) obtains the highest response since the white rectangle covers a blue region which conforms to the blue grass texton t. Therefore the algorithm proposed by Shotton et al. learns that cow

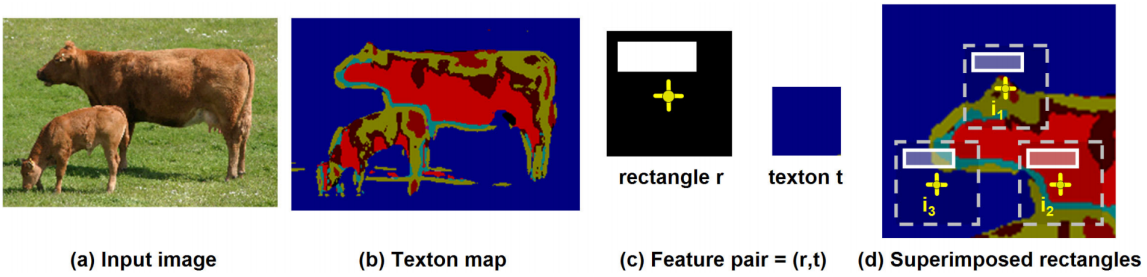


Figure 2.15: An example of the approach of Shotton et al. [75] to incorporate context within detection. For an image (a) they obtain a texton map (b). Based on a feature pair of a white rectangle r and a *grass* texton patch t (c), they obtain feature responses based on locations i. In this example i_1 (d) obtains the highest response since the white rectangle covers a blue region which conforms to the blue grass texton t. using this approach the algorithm learns that cow pixels tend to be surrounded by grass.

pixels tend to be surrounded by grass, improving the segmentation of a cow.

Another approach to using context by Kumar et al. [48] combines local and global contexts in a hierarchical field framework. In this hierarchy, the local context captures *short range* interaction at a pixel level, similar to the approach of Shotton et al., while the global context captures a *long range* of interactions where groups of pixels that correspond to regions or objects are modelled with respect to one another. Kumar et al. show that this unified approach of modelling context at different levels is beneficial in tackling problems of image labelling and contextual object detection.

There has been various other approaches where scene context has been used in terms of the relationship between the scene and the object [65, 87, 66]. In a related approach, Russell et al. [69] perform object recognition by obtaining a 2D scene *gist* by computing global statistics and obtaining representations of an image, providing the context of an object. A *gist*, defined by [31], generally refers to an abstract representation of the scene that spontaneously activates memory representations of scene categories. Therefore, in this work by finding the *gist* of a target image, they find different objects in the target image based on other matched images that have a similar *gist*. As illustrated in Figure 2.16, given an input image (a), Russell et al. use the *gist* feature introduced by Oliva et al. [61] to find matching images, presented in (b), that have a similar *gist* as the input image. Using a probabilistic model, they transfer object labels from the best matching images onto the input image to detect objects within it (c).

A similar holistic approach is used by Li et al. [52] where semantic scene context for object detection is exploited for event classification, where object and scene categorisations are integrated. Spatial context along with co-occurrence of objects are used as contexts

(a) Input image (b) Images with similar scene configuration (c) Output image with object labels transferred

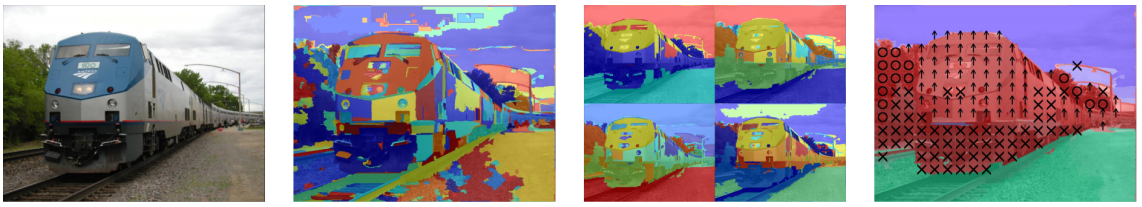
Figure 2.16: The approach of Russell et al. [69] for *gist* based contextual object recognition of a target image (a). Using scene *gist* they match the target image to other images that have a similar *gist* (b). They then use the labels in the best matching images and transfer them onto the target image.

in approaches by Galleguillos et al. [32] and Heitz et al. [38]. These work also follow the trend of previous approaches, the relative location between objects are modelled using pairwise features.

Geometric scene structures such as surface and viewpoint have also been shown to provide suitable contextual information for improving object recognition. Liu et al. [54] integrate a multi-view object representation with a unified spatio-temporal context model. The spatial context features include surface and viewpoint while the temporal context captures probability maps and local object trajectory predictions as prior probabilities. Another approach is to model factors such as the interdependence of objects, surface and camera viewpoint and to use them in an iterative fashion in order to refine each other as presented by Hoiem et al. [39].

In their earlier work, Hoiem et al. [40] provide geometric context by estimating scene structures from a single image. In this approach, as illustrated in Figure 2.17, given an input image (a), they obtain superpixels (b) and create multiple potential groupings of the superpixels. Given these groupings they then classify and label each image pixel, illustrated in figure 2.17 (d), as being (i) part of the ground plane, (ii) belonging to a surface that sticks up from the ground, e.g. a building or (iii) being part of the sky. In the work by Divvala et al. [27], aforementioned types of contexts in addition to new ones such as geographic context are combined within images for object detection.

In literature however, contextual information has been mostly used for filtering the detections prior to tracking or filtering the tracks post tracking, rather than influencing the linking during the tracking process itself. For example, Stalder et al. [77] focus on filtering tracks using contextual information such as a viewpoint filter, foreground filter and trajectory-like filter. In the next section we present related work where event context



(a) Input (b) Superpixels (c) Multiple Hypotheses (d) Geometric Labels

Figure 2.17: Geometric context labels proposed by Hoiem et al. [40]. For an input image (a) they obtain superpixels (b) in which they obtain carious groupings for (c). In (d) they then assign one of three labels to each image pixel, (i) being part of the ground plane, (ii) belonging to a surface that sticks up from the ground or (iii) being part of the sky.

has been used within the tracking process.

2.3.2 Event Context

Very few works have combined tracking and event detection. Li et al. [53] incorporate object-level spatio-temporal relationships, as context using a dynamic MRF to improve the inference of object categories and additionally improve tracking. These relationships provide a notion of events. This approach has three key concepts; (i) spatial relationships are incorporated between object categories such that co-inference enhances accuracy, (ii) temporal context is used to gather object evidence and to track objects continuously and (iii) *key objects* (such as humans) are robustly detected using other state-of-the-art approaches to reduce inference space for other objects and to improve the recognition.

These key concepts are illustrated in Figure 2.18, where given a video sequence Li et al. initially obtain key object detections (humans). Using a dynamic MRF, rather than using nodes to represent a pixel or a superpixel as is common in other contextual object recognition techniques, each node represents a hypothetical object in a single frame. Spatial and temporal relationships are modelled by intra-frame and inter-frame edges between object nodes respectively. To avoid building excessive false hypothetical object nodes within the MRF, the detected key objects provide contextual guidance for finding other objects.

In another approach, Wang et al. [93] join pedestrian tracking and event detection into a single optimisation problem. Their events describe human motion with respect to the viewpoint (left, right, away or towards the camera) which are extrinsic rather than intrinsic events, i.e. events are defined with respect to the viewpoint rather than the actors. Using a maximum a posteriori (MAP) optimisation they perform simultaneous tracking and event detection by means of Monte Carlo sampling similar to [41, 29, 35].

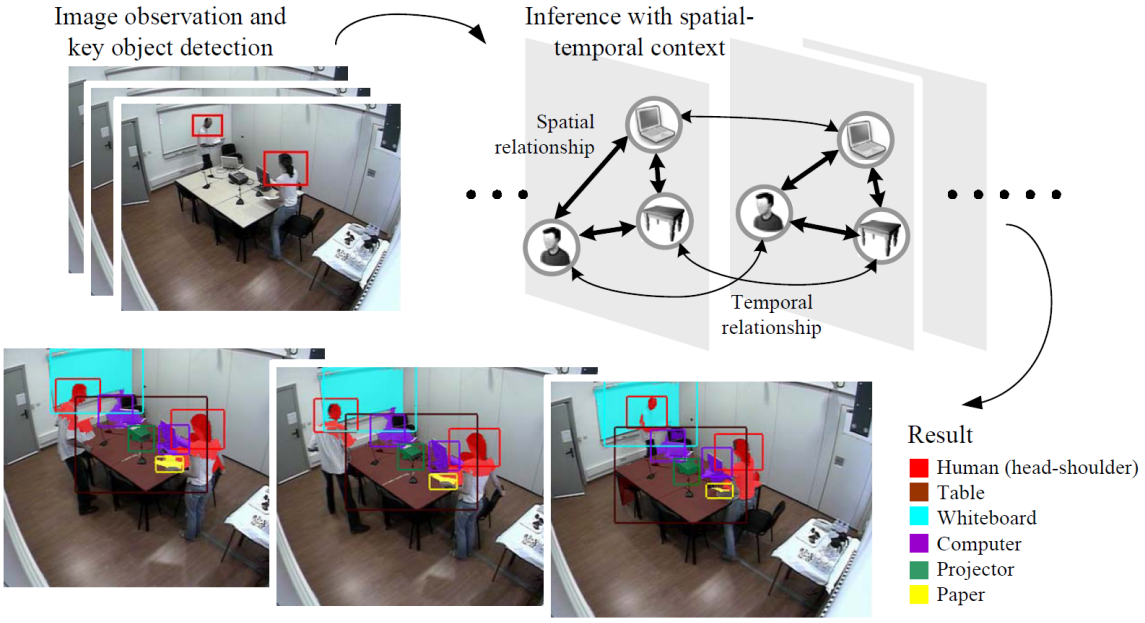


Figure 2.18: Object recognition and tracking using spatio-temporal context by Li et al. [53]

In a similar approach Choi et al. [17] jointly perform person tracking and activity recognition under a unified framework. The different types of person activities, namely *atomic*, *interaction* and *collective*, are modelled within a three level hierarchical structure. Their activity model does not solely rely on person tracklets and additionally takes as input external features such as appearance properties and context descriptors at various levels of the hierarchy. While they demonstrate the obtained person tracks are improved as a result of incorporating activities within their unified framework, their is no indication that the activities are also subsequently improved.

The work by Barbu et al, [5] focuses on simultaneous object detection, tracking, and event recognition. In their framework the tracking aspect is conditioned to event models. However event recognition and tracking are not optimised jointly; tracking requires event knowledge, and only running the optimisation for each event class allows to compare the results of event choice.

Discussion

Whether context is used in detection or tracking, it significantly improves the quality of detections and tracks produced. This is particularly the case for event based context which has gained attention in recent literature. However, each tracker incorporating this type of context is limited to one or both of the following features; (i) the events that are used are relative to the camera and do not contain any notion of interaction between an entity and

an object, and (ii) tracking and event analysis are not performed jointly, that is, tracks are not improved by the events that they produce and vice versa.

2.4 Conclusions

In this chapter we presented related work in the areas of carried object detection, tracking and context based detection and tracking. In each case we described the strengths and limitations of the related approaches. Here we present a conclusion on how the presented related work affected the design of our three main frameworks.

The related work on carried object detection showed that model based approaches are limited to only a certain number of objects types. It is therefore important for our detector to detect a generic class of objects, thereby leading to using geometric shape models at the core of our geometric carried object detector. The knowledge of protruding regions and areas belonging to a person provide a great indication on the location of the object. We therefore also incorporate this knowledge within our detector, however, we do not heavily rely on them.

Based on the related work on tracking, both our spatial consistency tracker and the tracker in our joint tracking and event analysis framework follow a similar approach to the Markov chain Monte Carlo data association algorithm by Oh et al. [59] where our optimisation uses moves to accomplish its goal. Since the discrete-continuous tracker by Andriyenko et al. [4] showed the importance of obtaining trajectories by considering both the discrete and continuous space, we initially construct tracklets using our spatial consistency tracker in the discrete space and finally form trajectories in the continuous space using the tracker in our joint tracking and event analysis framework, both of which take context into account. Moreover, the optimisation of each our trackers is posed as a *maximum a posteriori* (MAP) optimisation similar to many of the related work.

While we have worked on incorporating scene context within tracking [85], in this thesis we primarily focus on event based context. We overcome the limitations of other event based contextual trackers by simultaneously performing tracking and event analysis jointly within one objective function, where our notion of events fully capture interactions between entities and objects. Similar to the *key objects* used in the work by Li et al. [53], we capture interactions between an object relative to *reference entities*.

Based on these concepts, in the following chapters we present our geometric carried object detector, our spatial consistency tracker and our joint tracking and event analysis framework.

Chapter 3

Geometric Carried Object Detector

3.1 Introduction

Detection and tracking of carried objects is an important component of vision systems whether these are surveillance systems that aim to detect events such as leaving, picking up or handing over luggage, or robots that learn to perform better in indoor environments by analysing events involving humans interacting with carried objects. Despite significant progress in object detection and tracking, the task of detecting and tracking carried objects well enough to be able to use them for activity analysis is still a challenging problem. This task is elusive due to the wide range of objects that can be carried by a person and the different ways in which carried objects relate to the person(s) interacting with it e.g. carrying, dropping, throwing or exchanging.

In this chapter we describe our novel geometric carried object detector. We generalise the definition of a carried object as any particular object that an entity has interacted with in the scene, therefore not being limited to only when it is carried. This detector is specifically designed to overcome many limitations of other state-of-the-art detectors, as outlined in Chapter 2, and uses geometric shape models to characterise carried objects. The key concept of this detector is to allow for the detection of a generic class of objects, regardless of their shape and structure, effectively removing the need to train specific object models.

In the following sections we present our geometric carried object detector. We initially describe the process of obtaining object boundaries followed by a cost function used to

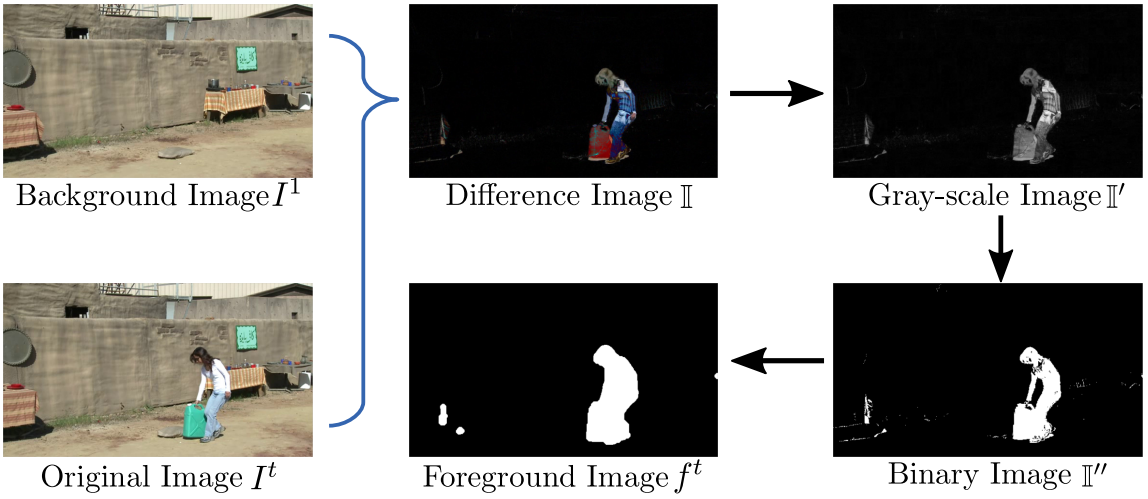


Figure 3.1: Foreground extraction process.

evaluate each boundary in measuring its suitability as an object. We finally describe the process of obtaining object detections based on the obtained boundaries.

3.2 Detection of Foreground and Protrusion

We consider a video $\mathcal{I}$ consisting of a time series of images $\mathcal{I} = \{I^1, ..., I^t, ..., I^N\}$ in which we represent their corresponding sequence of foreground masks as $\mathcal{F} = \{f^1, ..., f^t, ...f^N\}$. Each foreground mask f^t is a binary image of ones and zeros where the zeros correspond to the background regions of the image and the ones correspond to foreground regions in the image, also indicating person and object silhouettes. As illustrated in Figure 3.1, to obtain each foreground mask f^t, we initially subtract the foreground's corresponding frame I^t from a single frame containing only the background (which we assume to be the first frame I^1) followed by taking the absolute value, resulting in a difference image $\mathbb{I} = |I^1\text{-}I^t|$.

We then obtain a gray-scale image $\mathbb{I}'$ from the difference image $\mathbb{I}$ by assigning each pixel at (i, j) in the gray-scale image, the maximum value of the corresponding pixels in each of the RGB channels of $\mathbb{I}$ i.e. $\mathbb{I}'_{ij} = \max(\mathbb{I}^r_{ij}, \mathbb{I}^g_{ij}, \mathbb{I}^b_{ij})$. A binary mask $\mathbb{I}''$ is then constructed by assigning $\mathbb{I}''_{ij} = 1$ if the value of $\mathbb{I}'_{ij}$ is greater than the mean intensity of $\mathbb{I}'$, and $\mathbb{I}''_{ij} = 0$ otherwise. This thresholding mostly removes noise from background motion. We apply various morphological operations [36] by initially filling any holes in the mask, followed by a closing operation to fill any remaining open regions on the boundary of the mask. Finally, we perform dilation to slightly increase the mask so that any edges on the person or object boundary are guaranteed to be on the foreground mask. As a result

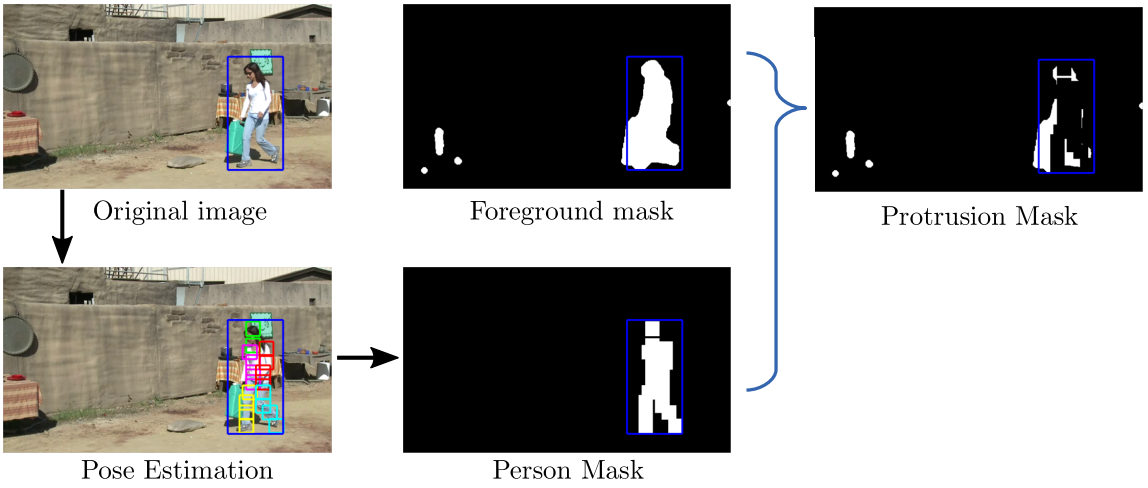


Figure 3.2: Process of obtaining the protrusion mask.

of applying the aforementioned stages on an image I^t, we obtain a foreground mask f^t illustrated in Figure 3.1.

In addition to foreground masks, we also obtain protrusion regions which rely on having person tracks. We therefore obtain person tracks by initially obtaining person detections by applying Felzenzwalb et al. [30] part-based object detector with the VOC release version 5 [34]. We then apply a state-of-the-art tracker by Pirsiavash et al. [63] on the person detections to obtain person tracks.

We obtain a protrusion mask for each detected person bounding box in a person track in two steps, as illustrated in Figure 3.2. First, we apply the articulated pose estimation code by Yang et al. [100] within each person bounding box where the size of each box has been slightly expanded. From this we obtain various smaller bounding boxes that are body part estimates inside the bounding box of the person. We define a person mask as the union of regions covered by these body part bounding boxes. We subtract the person mask from the foreground mask and consider any remaining regions of the subtracted mask as regions in the protrusion mask.

In the next section we formally define and describe our method for carried object detection which makes use of the obtained foreground, person tracks and protrusion masks.

3.3 Carried Object Detection

Edges play an important role in our carried object detector as the detector primarily relies on edges to obtain carried object detections. However, edges in their original form may contain too much noise, are jittery and are not straight forward to work with. To avoid

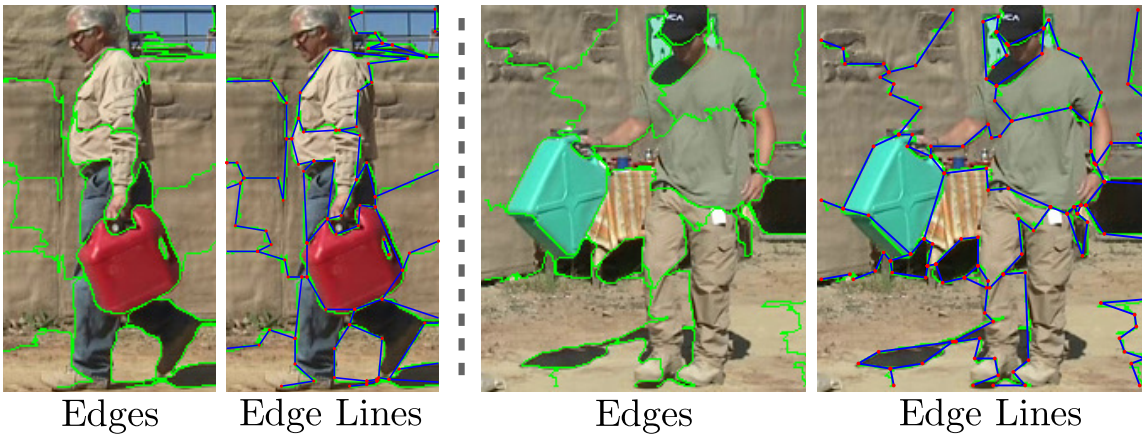


Figure 3.3: Two examples of obtaining edge lines from edges. In each case the left image shows the edges obtained by using the Quick Shift algorithm in green. The right image shows edge lines in blue where a red dot is the start or end of the edge line.

these issues we convert edges to edge lines. We initially obtain edges by either using an edge detector, e.g. Canny [15], or a segmentation algorithm that provides boundaries, e.g. Quick Shift [91]. Edge lines are then obtained by applying Kovesi's *edgelink* function [46] version 2007, which links connected edge points to form lines. If the edge point hits a junction (an edge point is connected to multiple edge points) or deviates too much from the line, the line is broken and a new line is created. Results of applying edge lines from edges are illustrated in Figure 3.3 for two examples. We now describe the formal description of our carried object detector on these edge lines.

Given a set of edge lines L, we represent the power set of these edge lines as $\mathcal{P}(L)$. We then denote $\mathbb{L} = \mathcal{P}(L)$ as the set of all possible permutations of edge lines in L. The goal of our carried object detector is to provide the subset of $\mathcal{L} \subseteq \mathbb{L}$, where each $l \in \mathcal{L}$ is a set of edge lines that can be ordered and linked into a chain defining the boundary of an object given a target shape model. The conformity of candidate chains to a target shape model is measured by exceeding a fixed threshold for a cost function. We consider two target models of shapes, namely *convex* and *elongated* objects. Rather than searching exhaustively through $\mathbb{L}$, we use an efficient level-wise mining method that approximates $\mathcal{L}$, generating most boundaries $l \in \mathcal{L}$ but not all $l \in \mathbb{L}$.

We represent the boundary b of a potential object as a polygon obtained from the chain of edges in l. For an l to be a member of $\mathcal{L}$, i.e. $l \in \mathcal{L}$, its boundary b must have a cost higher than a certain threshold. This cost is obtained by normalising the cost $C(b)$ over all detections and is based on various other costs, which we would like to maximise individually, that are defined by the following:

$$\mathcal{C}(b) \;=\; \mathcal{C}_g(b, \Theta_{\mathrm{g}})\ \mathcal{C}_c(b)\ \mathcal{C}_p(b) \tag{3.1}$$

The first factor, $\mathcal{C}_g(b, \Theta_{\mathrm{g}})$, measures the conformance of the edge lines l of a boundary b to a given shape model Θ_{g}. In other words, the polygon of the edge lines l representing an object boundary must conform to a certain geometric shape model. In Figure 3.4, the set of edge lines that have a more convex shape obtain a higher and better cost. The geometric shape property is described in section 3.5.

The second factor, $\mathcal{C}_c(b)$, calculates the *connectivity* of the edge lines forming the edge chain. In an ideal case, for a chain covering the boundary of an object, all edge lines in l meet, creating a closed chain. $\mathcal{C}_c(b)$ calculates a connectivity cost by taking the ratio of the length of the edge lines in l over the length of its connected form which includes the length of both interpolated and non-interpolated edge lines. If the edge chain is closed, a ratio of one will be obtained and if not, the ratio will be closer to, but not less than, 0.5. A high and low cost edge chain based on connectivity is illustrated in Figure 3.4.

The third factor, $\mathcal{C}_p(b)$, measures the proportion of an object's boundary that overlaps with the protrusion mask described in Section 3.2. This measure is beneficial as protruding regions are more likely to only belong to an object, whereas non-protruding regions may belong to either an object or a person region. In Figure 3.4, the white mask represents protrusion and the red lines represent the chain of edge lines defining the boundary of an object. We can observe that in the top case the red chain overlaps more with the protrusion region and obtains a better cost, while the bottom case overlaps less and obtains a lower cost.

The protrusion measure $\mathcal{C}_p(b)$ is obtained by Equation 3.2, calculating the ratio of the area of intersection between the object boundary and the protrusion mask, over the area of the boundary of the object. More specifically, we represent the object boundary as a mask. This mask is obtained from a polygon that is created from the object's edge chain. The process of obtaining the object mask is described in section 3.4. Therefore to obtain the intersection area, we find the are of the overlapping mask between the object and the protrusion masks.

$$\mathcal{C}_p(b) \;=\; \frac{\text{Area}(\text{object} \cap \text{protrusion})}{\text{Area}(\text{object})} \tag{3.2}$$

After obtaining all boundaries b in $\mathcal{L}$ and their corresponding costs from $\mathcal{C}(b)$ in Equation 3.1, we normalise all the costs to a range between 0.01 and 0.99, and treat them

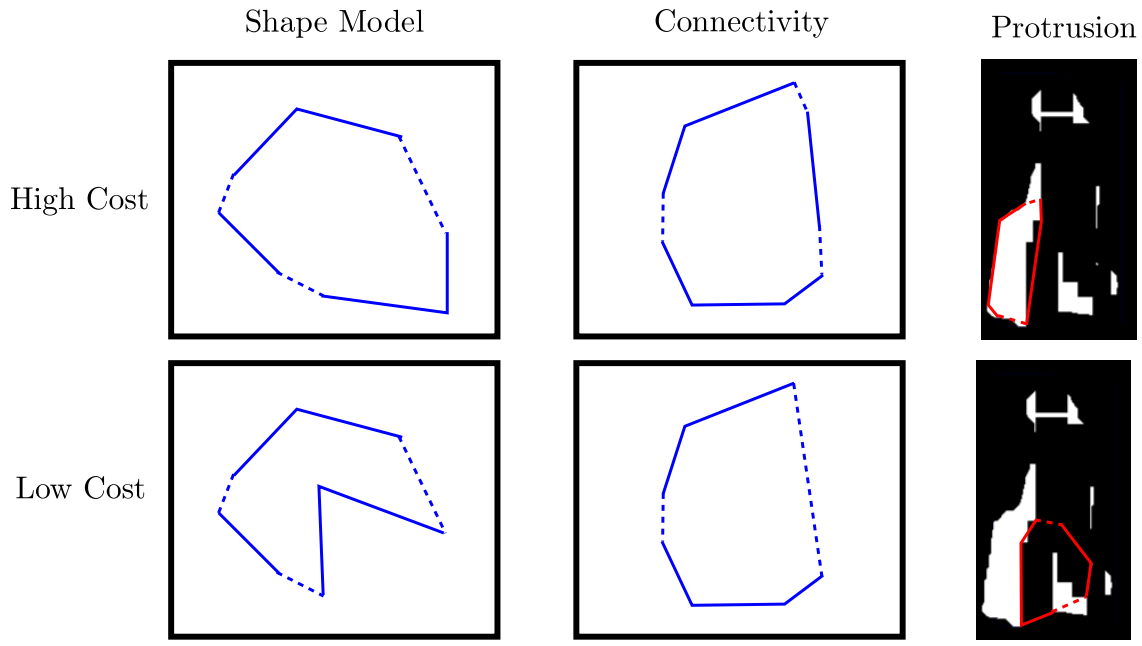


Figure 3.4: High cost (good) and low cost (bad) examples of each of the factors in Equation 3.1. Solid lines represent edge lines and dotted lines represent the interpolated edge lines, together forming a chain that defines the polygon of an object and its boundary. The first column represents the cost of conforming to a shape model where the convex polygon (top) obtains a better cost while the concave polygon obtains a less and poor cost. The second column measure the connectivity cost of the object function. Since the top polygon has more edge lines covering the boundary of the polygon rather than interpolated edge lines, it obtains a better cost. In the bottom case however the interpolated edge lines are much longer, leading to a low cost. The third column represents the protrusion cost in the objective function. The top example obtains a better cost as the boundary of the object covers more protrusion regions (white pixels), while the bottom case covers less and subsequently obtains a lower cost.

as detection likelihoods in future computations. The cost of each term in Equation 3.1 is not weighted and the product follows a linear distribution.

In the next section we describe the process of obtaining the ordering of the edge chains from the set of edge lines l and their corresponding polygon.

3.4 Object Mask

To measure the conformance of a boundary b to a shape model or its measure of protrusion from a person, i.e. first and third factors in Equation 3.1, we require an estimate of the object region in terms of the edge lines l that define the boundary b. We represent this object region as a mask which is obtained in three steps, the creation of (i) an edge chain

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

Figure 3.5: Creating edge chains by finding the best ordering of its edges.

from edges in l (ii) a polygon from the edge chain and (iii) a convex hull from the polygon.

3.4.1 Edge Chain

In the first step, we must construct an edge chain that defines the boundary of the object which will be later used to create a polygon. As a result, given a set of edge lines l, the creation of edge chains is constrained by the ordering of edge lines that form a polygon conforming to a certain shape model. Although the ordering of elements already present in l gives a notion of how to connect the edges together to form the chain, this ordering of edge lines does not facilitate the ordering in which the edge line's start and end points should be connected to each other. Therefore, the main goal of edge chain creation is to find the best order of edge lines in l to be connected with respect to their start and end points such that the resulting polygon would produce high conformity to a geometric shape model.

To solve the problem of finding the best order of edge lines, the problem can be posed as finding the order in which various points (start and end points of each edge line) can be connected to each other, with the constraint that some points must be connected to each other (due to their edge connections) and the resulting ordering produces high conformity to a shape model. Here we describe the process of finding this ordering for a specific geometric shape model that prefers highly convex polygons (this geometric shape model and others are described in Section 3.5).

Figure 3.5 illustrates the process in which we construct our edge chains. Given a set of

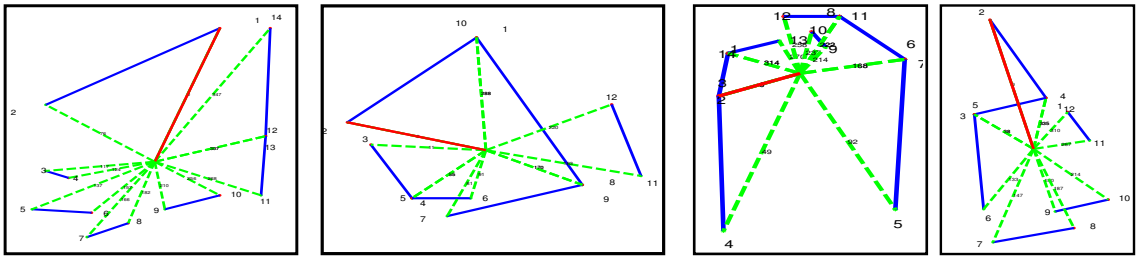


Figure 3.6: Additional examples of edge chain ordering.

edge lines, blue lines in (a), the first step to finding a polygon is to find the centre of all the start and end edge points. We then draw a line from this centre point to the start point of the first edge line in l, as illustrated with a red line in (b). In the second step we draw a line from each edge point to the centre point as displayed with green dotted lines in (c). Using the red line as reference, we obtain the angle between each green line to the red line. All of the obtained angles are displayed on each green line in (c). In the final step we enforce the constraint of building a highly convex shape edge chain. The ordering of the edge lines is based on their rotational position relative to the obtained centre point. As a result the edge line with a start or end point that has the shortest angle is the first edge in the chain followed by the next edge point with the shortest angle and similarly for other edges. The result of this ordering can be seen in (d) where each edge point is assigned a number in yellow indicating their order in the chain. Numbers are slightly displaced to avoid overlap.

This approach to ordering produces an edge chain more naturally defining a boundary rather than allowing edge points to be connected to points on the other side of the centre point and then coming back. By simply linearly connecting each of the points based on their number, an edge chain is obtained. In Figure 3.6 various other examples of ordering is presented.

In the next section we describe how the ordered chain of edge lines is used to produce a polygon and a convex hull representing the object mask.

3.4.2 Polygon and Convex Hull

In order to calculate various measures based on the shape of an object, we must define an object region. As illustrated in Figure 3.7, we obtain this region by linearly interpolating the points in the edge chain ordering, resulting in a polygon. We then obtain a mask of this polygon and use it to represent our object mask. Similarly we obtain a convex hull mask from the polygon using the standard convexity measure and approach described by Zunic et. al [105].

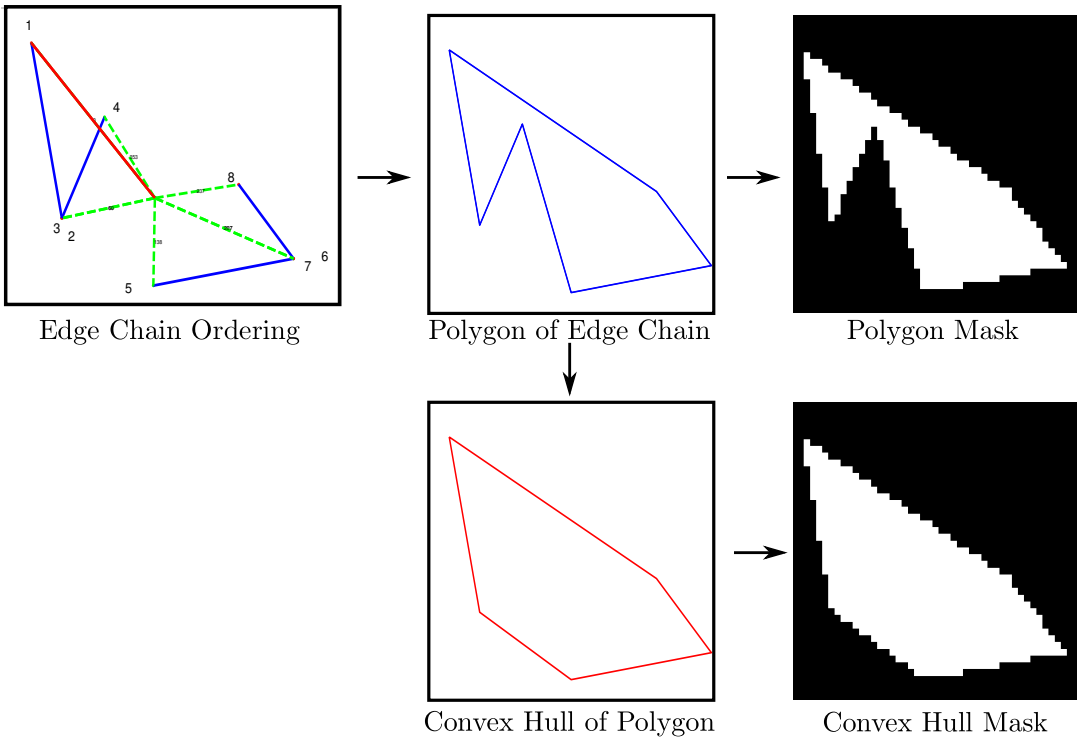


Figure 3.7: Example of an ordered set of edge line, the corresponding polygon leading to an object mask with a non convex shape and the convex hull of the polygon.

In the next section, we describe the geometric shape property term $\mathcal{C}_g(b, \Theta_g)$ of Equation 3.1 which makes use of the polygon and convex hull to measure the conformity of an object mask to a shape model.

3.5 Geometric Shape Properties

We regard an object boundary b as more likely to be a carried object if the shape of the detection region (defined by the object mask) conforms to any of the pre-defined generic geometric shape properties (Figure 3.8). The term $\mathcal{C}_g(b, \Theta_g)$ in equation 3.1 measures this conformity with respect to a single shape model in the set of geometric shape models Θ_g. The carried object detector runs for each of the shape models in Θ_g independently providing a set of object detections $\mathcal{L}$ for each case.

In this work we describe two types of shape models, namely *convex* and *elongated*. While we present examples of both models, we primarily apply our framework on the *convex* shape model as it covers the most variety of carried objects. We consider a convex shape model with parameter $\theta_{\text{con}} \in \Theta_g$ and an elongated shape model with parameter

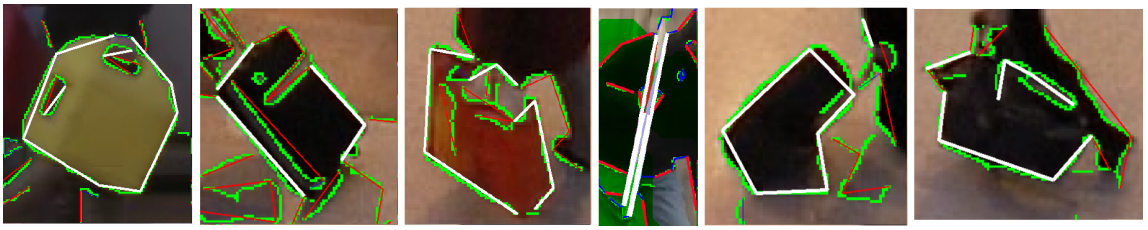


Figure 3.8: Examples of using geometric shape models for carried object detection. Green segments represent edges from the Canny edge detector and the white chain lines represent the convex/elongated objects found in the level-wise mining procedure.

$\theta_{\text{el}} \in \Theta_{\text{g}}$. The choice of these shape models are based on the fact that most carried objects have a shape that is approximately convex (e.g. briefcases, backpacks, boxes and petrol cans) and many are elongated (e.g. objects with an elongated part such as shovels, rifles and brooms). These two types of shape models allow us to cover a wide range of object types.

We evaluate the cost of a boundary for a convex model $\mathcal{C}_g(b, \theta_{\text{con}})$ by computing a degree of convexity based on the areas of the polygon of an object boundary b and the polygon's convex hull (*hull*), which were described in section 3.4.2:

$$\mathcal{C}_g(b, \theta_{\text{con}}) \quad = \quad \frac{\text{Area}(b)}{\text{Area}(\textit{hull})} \tag{3.3}$$

To evaluate the cost of a boundary for an elongated model $\mathcal{C}_g(b, \theta_{\text{el}})$, we compute a degree of parallelism between candidate sets of edge lines l which can be partitioned into two non-overlapping proximal groups of approximately co-linear edge lines, g_i and g_j. We initially describe the method of obtaining a group g, followed by our elongated measure which uses parallelism.

In order to obtain an approximately co-linear group of edge lines g, we must use a suitable measure that calculates co-linearity. In the literature, there have been various approaches to calculating a co-linearity measure between two edge lines [56, 88] which primarily focus on combining angle and distance parameters between the edge lines.

In this work however, for every line $E1$ and $E2$ in a chain of edges forming the boundary b, we define their co-linearity measure as a sum of angles between them. As illustrated in Figure 3.9, we calculate each angle ψ between two lines. The first line is created by connecting the starting point $P1$ on one edge line to either the first or second point on the second edge line ($P1$ or $P2$). This line is illustrated as red in Figure 3.9. We assign the second line as the second edge line. Since there are two starting points (one on

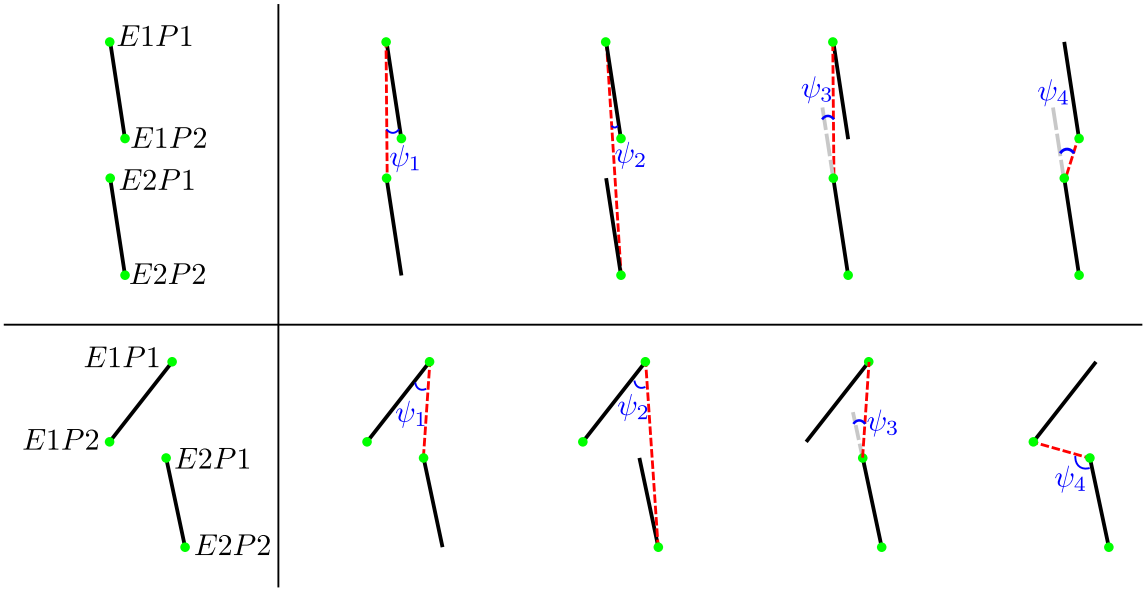


Figure 3.9: The four different angles ψ used to obtain a co-linearity measure between two edge lines $E1$ and $E2$. Each edge line has a start point $P1$ and an end point $P2$. Red lines are used to connect two green edge point to each other. An angle ψ is obtained between the red edge line and a black edge line between two green points.

each edge line) and there are two points on each edge line to connect to, we obtain four combinations of line pairs leading to four angles. If an angle is greater than 90 degrees, we subtract it from 180 (illustrated using a gray line). After taking the sum of all angles, if the sum is less than a threshold and closer to zero, we consider the edge lines to be highly co-linear and use them as a co-linear group of edge lines g.

To obtain a final co-linearity measure for a chain of edge lines that has more than two edges, we take the average co-linearity measure between every pair of consecutive edge lines in the chain. We use the same co-linearity threshold to define a group g.

After obtaining various co-linear groups, by using parallel and proximal properties, we define and calculate our elongatedness measure as the following product:

$$\mathcal{C}_g(b, \theta_{\text{el}}) \quad = \quad \mathcal{N}(Z(g_1, g_2)|\theta_a)\ \ \mathcal{N}(D_1(g_1, g_2)|\theta_d) \tag{3.4}$$

In Equation 3.4, $\mathcal{N}(Z(g_1, g_2)|\theta_a)$ measures the degree of parallelism using the normal distribution $\mathcal{N}$ with a model θ_a based on the angle between the two co-linear groups g_1 and g_2 calculated by function Z. The second term in the product, $\mathcal{N}(D_1(g_1, g_2)|\theta_d)$, measures the closeness between the two groups by using the normal distribution $\mathcal{N}$ with a model θ_d in terms of their shortest Euclidean distance to each other, calculated by function D_1.

Therefore by using Equation 3.4, co-linear groups of edge lines that are highly parallel and in close proximity obtain a higher and better cost for $\mathcal{C}_g(b, \theta_{\text{el}})$.

To find object boundaries $b \in \mathcal{L}$ for both convex and elongated shape models using the aforementioned convexity and elongatedness measures, we must search through various combinations of edges $l \in \mathbb{L}$. However, if the size of L is very large, it becomes computationally expensive to perform an exhaustive search through each $l \in \mathbb{L}$ to find $\mathcal{L}$. Therefore, in the next section we describe a heuristic level-wise mining approach to search for and approximate the set $\mathcal{L}$ of candidate edge chains, avoiding the creation of $\mathbb{L}$.

3.6 Edge-based Level-wise Mining

In order to avoid the computation of all possible combinations of edges to obtain $\mathbb{L}$, we require a search procedure to build $\mathcal{L}$ without computing all elements in $\mathbb{L}$. This search procedure must construct various length edge chains $l \in \mathcal{L}$ and must do so within a significantly reduced search space of constructing $\mathcal{L}$ when compared to building the set of all edges $\mathbb{L}$. We therefore incorporate an edge-based level-wise mining search procedure which constructs edge chains l at each level of this procedure. The constructed edge chains are then merged to create longer edge chains in future levels. This approach approximates the process of obtaining $\mathcal{L} \subseteq \mathbb{L}$ as it produces most of all edge chains $l \in \mathcal{L}$ and $l \in \mathbb{L}$ while removing edge chains where $l \notin \mathcal{L}$ and $l \in \mathbb{L}$ and avoiding any future combinations of them in the level-wise mining search space.

This notion of using item sets of each level to be merged forming those of the next level was initially introduced by Agrawal et al. [1] and further developed and used in a variety of areas [55, 9, 81]. In this section we describe our approach of using the level-wise mining technique and applying it on an edge-based approach. We start by describing its general framework followed by its procedure and finally, the constraints that each l must satisfy to be accepted in $\mathcal{L}$.

Framework: At each level k in our level-wise mining procedure, we create a set $\mathcal{L}_k$ containing chains of edges l, where each $l \in \mathcal{L}_k$ has a length of k edge lines. Each edge line subset l at a level k is constructed by taking two distinct candidate $k-1$ subsets from the set $\mathcal{L}_{k-1}$ and merging them if they share $k-2$ edge lines. The new subset l is accepted as a level k candidate set if it satisfies certain constraints, e.g. highly conforming to a geometric shape model ($\mathcal{C}_g(b, \Theta_{\text{g}})$ in equation 3.1). If not, it will not be added to the $\mathcal{L}_k$ set and cannot take part in the creation of $\mathcal{L}_{k+1}$.

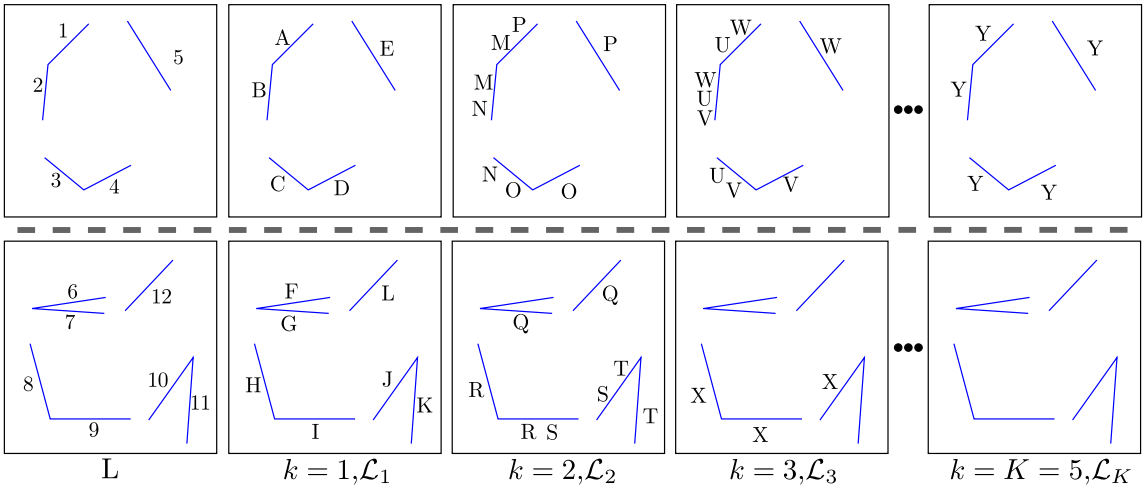


Figure 3.10: The level-wise mining procedure applied on two sets of edges from L. The first row showcases finding a long chain of convex set of edges l as a potential object. The bottom row showcases a set of edge lines that do not produce a long chain of edges as the level-wise mining progresses to each level k due to constraints. The effects of the constraints can be seen between each level. In the first column a number represents a specific edge line. In all other columns a letter represents an edge set ID where an edge line near a letter belongs to that specific edge set. An edge can be a member of multiple sets.

Procedure: Illustrated in Figure 3.10, we start the level-wise mining process at level $k = 1$, where we initially assign the given set of all edge lines L to $\mathcal{L}_1$. To construct $\mathcal{L}_2$, at $k = 2$, we assume every pair of edge lines from $\mathcal{L}_1$ share a common edge and merge them as one set in $\mathcal{L}_2$, if the constraints are satisfied. To construct any $\mathcal{L}_k$ where $k \geq 3$, we follow the generic process previously outlined for the level-wise mining procedure at each level k.

After the level-wise mining process has completed running for all k levels, where $1 \leq k \leq \mathcal{K}$, we construct the set of all possible edge chains as $\mathcal{L} = \{\mathcal{L}_k \cup ... \cup \mathcal{L}_\mathcal{K}\}$ for any $k \geq \lambda_k$. The value λ_k is a set number representing the shortest allowed length an edge chain l can have to be considered an object. If any two edge chains $l_i \in \mathcal{L}$ and $l_j \in \mathcal{L}$ exist such that $l_i \subset l_j$, the edge chain l_i is removed to avoid repetitive shorter detections in $\mathcal{L}$.

Constraints: As previously mentioned, each l must satisfy certain constraints to be accepted in a $\mathcal{L}_k$. These constraints are namely (i) distance, (ii) angle and (iii) the geometric shape model described in Section 3.5. For the distance constraint, if two edge lines are to be considered for merging, the shortest distance between the two lines has to be less than a threshold (less than 20% of the human height). For the angle constraint, the angle between the two edges must not be very sharp and has to be larger than a certain angle

threshold (< 30). For the shape model constraint, the shape of l, i.e. the polygon, must highly conform to the shape model ($> 95\%$ convexity).

As the number of combinations (possible merges) in each level of the level-wise mining can become extremely large, the distance and angle constraints remove a very large combination of edges that are either too far away from each other to be on the same object or are unlikely to shape the boundary of the object due to their extreme sharp angles. Removing these combinations early in the level-wise mining process greatly increases speed and efficiency in later levels. After $k \geq 3$ however, there is no need to enforce the distance and angle constraints as they have already been enforced in level $k = 2$, in which their effect will continue on in future levels of the level-wise mining procedure. Therefore the distance and angle constraints are only enforced in level $k = 2$ and are not used in future levels, resulting only in a single constraint being enforced in future levels, the geometric shape model.

In Figure 3.10 we illustrate the level-wise mining procedure for two cases. The top row showcases the first where the level-wise mining process for each k is presented, resulting in finding a long chain of convex set of edge lines l as a potential object. The bottom row showcases the second case which results in this process not finding a suitable set at $k = \mathcal{K}$ due to many edge lines not satisfying constraints. In each case, the first column represents the set of edges from L, where the level-wise mining is being applied to only the edge lines in a particular row. A number in the first column indicates a specific edge line. This number will be used to refer to specific edge lines from hereafter. In all other columns a letter refers to the ID of a specific edge set l. From hereafter, a subset ID is represented in bold, e.g. **T**, to avoid confusion with other notations in this thesis. It must be noted that the alphabetical ordering of the letters has no meaning and they could have been randomly assigned. Therefore, if an edge line has an edge set ID next to it, it is a member of that specific edge set. Moreover, a single edge line can be a member of multiple sets. We can observe that at $k = 1$, each edge line is assigned an l ID. At level $k = 2$ many merges are accepted as they satisfy the three aforementioned constraints. In the top row, we can observe the set l with an ID of **M** being created by merging the sets **A** and **B** together. As a result, set ID **M** contains edge lines 1 and 2.

There are also a few cases where constraints are not satisfied. Edge set IDs **F** and **G** are not merged at $k = 2$ as the angle between them is too sharp. Edge set IDs **D** and **E** are also not merged as the distance between them is too large. In level $k = 3$, we can observe the merging of length two edge chains l from $k = 2$, producing length three edge chains. For example, in the bottom row, edge sets with IDs **R** and **S** at $k = 2$ are merged since they share a $k - 2$ edge, i.e. edge number 9, resulting in edge set ID **X** at $k = 3$. However, the

merging is not allowed for edge set IDs **S** and **T** as their merged chain does not satisfy the shape model constraint of having a convex shape.

As previously described, the level-wise mining process continues until it reaches $k = \mathcal{K}$ and outputs an $\mathcal{L}$ by taking the union of all $\mathcal{L}_k$ where $k \geq \lambda_k$. In Figure 3.10 we can observe that at the end of the level-wise mining process, the last column for the top row, a length five edge chain l is found while no suitable edge chains for the bottom case is created.

In the next section we describe how we remove certain edge lines from the set of all edge lines L, reducing the number of combinations in the level-wise mining.

3.7 Reduction of Edge Lines

In section 3.3, we described our carried object detector where it starts with a given set of edge lines L. It is natural however to obtain an extremely large number of edge lines from the aforementioned techniques. The size of this set of edge lines, L, will directly affect the computational speed of the level-wise mining procedure described in section 3.6. As a result, the larger L is for each frame of a video, the more combinations the level-wise mining has to go through.

Therefore, it will be very beneficial to remove edge lines from the set L that are most likely not on an object boundary. This reduction will greatly reduce the search space in the level-wise mining, leading to an increase in computationally efficiency and less false positive chains l.

In order to apply the carried object detector and to reduce the number of edge lines, the carried object detector is run on two types of regions, (i) the foreground region a person is covering and (ii) all other connected foreground regions not covered by a person. The following approaches are then used to reduce the number of edge lines in L, where examples of each are illustrated in Figures 3.11 and 3.12. The approaches that require a person are only applied on the first case where the foreground region is covered by a person. The person's bounding box is also slightly extended to cover potential protrusion cases.

Edge Enhancement: One of the most important types of edges that need to be removed are edges that do not form any particular boundary and are inside boundaries themselves. Therefore to avoid including these types of edges in the level-wise mining process, before running an edge detector, we apply the *Image Edge Enhancing Coherence Filter Toolbox*, developed by Kroon et al. [47], on any image region we want to find a carried object. This toolbox performs anisotropic non-linear diffusion filtering which reduces the image

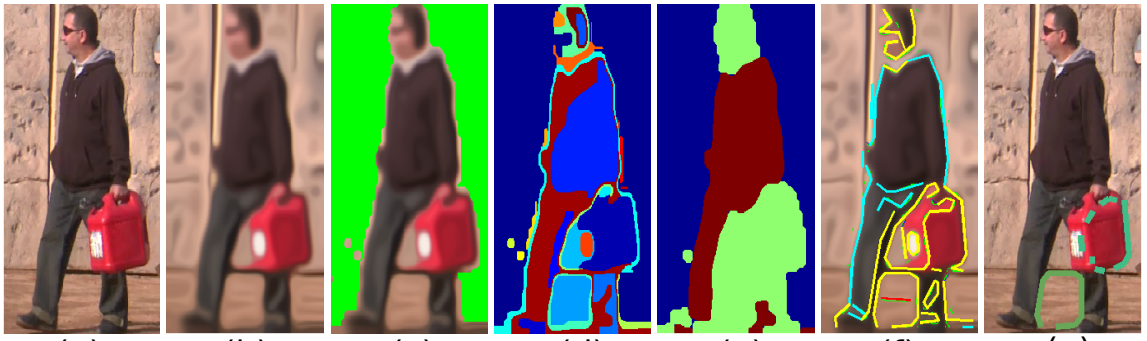


Figure 3.11: The process of obtaining candidate carried object detections. (a) We first obtain the image corresponding to the person detection; (b) We then apply the Edge Enhancing [47] method to enhance edges corresponding to natural boundaries; (c) We apply foreground extraction on *b* (background shown in green); (d) We apply colour based segmentation to *c*; (e) We identify the two largest segments (given in red) in *d*, which tend to correspond to regions on the person. The carried object is more likely to be present in the non-person regions (shown in green); (f) Using the regions identified in *e*, many of the line segments belonging to the person are removed (coloured with cyan); (g) The result of applying level-wise mining to the remaining edges (coloured yellow in *f*) to obtain candidate carried object regions (coloured in green).

noise (edges) within boundaries, while preserving the outer edges of the boundaries and additionally enhancing these edges by smoothing along them.

As a result of applying this approach, when edge extraction is run after this stage, we obtain significantly less edges within boundaries of carried objects and stronger edges defining the same boundaries. For a sample image Figure 3.11 (a), the affects of applying the edge enhancement technique is illustrated in Figure 3.11 (b). It can be observed that regions with more natural boundaries are present while regions within boundaries or with no specific boundary have been smoothed out by the filter.

Person Edges: Another edge type that contributes towards the majority of edges in L are person edges. These edges, whether they are on or around the boundary of the person region, are the main source of false positive chains of edges. Moreover, they also require a much larger computation time in the level-wise mining procedure compared to the edges on the carried object which are much fewer in number. Therefore, it would be very beneficial to remove these person edges from L in a conservative manner.

To accomplish this, we initially remove the background region from the edge enhanced image, as illustrated in Figure 3.11 (c). We then perform colour segmentation by applying a graph cut based algorithm developed by Boykov et al. [11] incorporating more recent

libraries of min-cut max-flow energy minimisation algorithms of [45, 10]. Based on the obtained colour segments, Figure 3.11 (d), we define the person region as the union of the top two largest segments, illustrated as the red region in Figure 3.11 (e). We have found that the two largest regions provide a reasonable estimate of finding the person region.

Two types of edges are removed from the obtained person region; (i) edges that are solely inside the person region (inside the red region of Figure 3.11 (d)) and (ii) edges that are on the boundary of the person region and the background region (on the shared boundary of the red and blue regions). Any edge that is on or within the boundary of the potential carried object region (green) is kept. The effect of identifying person edges is illustrated in Figure 3.11 (f) where removed person edge lines and kept edge lines are displayed in cyan and yellow respectively.

Person Parts: Although the method of finding person edges may remove a large number of edge lines from the set L, there may be certain person body parts that are not large enough segments to be removed. Certain parts may also highly follow the geometric shape models outlined in section 3.5, such as the head of a person which is highly convex and is sometimes detected as a carried object.

Since we run a pose estimator as part of the process to obtain a protrusion region, as described in Section 3.2, the location of the person's body parts are known. In our approach, we removed any edges that are on the *head* body part as the pose estimator was reasonably accurate in finding it. However, we found that it was not accurate enough to find the feet of the person. Accurately finding the feet would have been beneficial as many edges on the feet region form false positives which may build up along the legs. If a pose estimator is highly accurate, the feet regions may also be removed, however, as any other body part region such as the legs or the torso may be covered by the carried object, the removal of edges on such regions may lead to removing edge lines on the carried object and should therefore be avoided.

Edge Length: Since certain textures or items in the image may produce edge lines with very short lengths, it would be very beneficial to remove these short edges lines to speed up the level wise mining process. However, due to the depth of the image the length of a very short edge line may vary. Therefore, for each edge line, we calculate a ratio based on the length of the edge line over the height of the person that the detector is running for. If this ratio is less than a certain threshold (less than 5% of the human height), that edge line will be removed from L.

Colour Contrast: Even though the edge enhancement step smooths and eventually leads to removal of edges within a boundary, certain strong edges may remain inside these boundaries. These strong edges are mostly due to the creases on shirts or trousers or caused by a shadow on a part of a region while the other part is bright. To avoid including such edge lines in L, for each edge line, we compute a similarity measure based on the colour profile of the two sides of the edge line. For each side, we obtain a rectangular region with a fixed width parallel to the edge line. We then obtain and concatenate the RGB values in the two rectangles on either side of the edge line. We then calculate the standard deviation of each concatenated colour channel and calculate a Euclidean norm based on the three colour channel values. If this value is less than a certain threshold (< 20), even though it is a strong edge, we remove this edge line as the colour profile of the two sides of the edge are similar, and it is therefore not on a true boundary and merely lies within one.

By applying the above approaches we significantly reduce the number of edge lines in L, enabling us to generate a smaller set of edge chains in the level-wise mining process, while significantly removing false positives and greatly improving computational efficiency. However, the main reason in using the aforementioned approaches in a pre-processing stage is to clean up the data (edge lines), in a conservative manner, before starting our detector. An example of this is illustrated in Figure 3.12 (e) where numerous short edge lines are removed, indicated by a blue colour. Figures 3.11 (g) and 3.12 (g) illustrate the final set of edge chains l obtained from the level-wise mining procedure. Here each set is represented by a different colour and may be overlapped by another set.

After obtaining all object boundaries by completing the level-wise mining process, we represent an object boundary b as an object detection d by fitting a minimum enclosing rectangle to the polygon of the object boundary. In Figure 3.13 we present sample boundaries obtained as a result of applying our geometric carried object detector. The images include examples for both convex and elongated shape models before they are converted into rectangular object detections. These boundaries are obtained from a variety of datasets and highlight the fact that our approach can localise the boundary of an object and subsequently its mask accurately, due to the direct use of edge lines. The last two images present examples for the elongated shape model while all other images are for the convex case. In Chapter 6 Section 6.4 we perform experiments to evaluate the quality of our obtained detections. These detections will be used in future chapters as part of the tracking process where we only focus on the convex shape model.

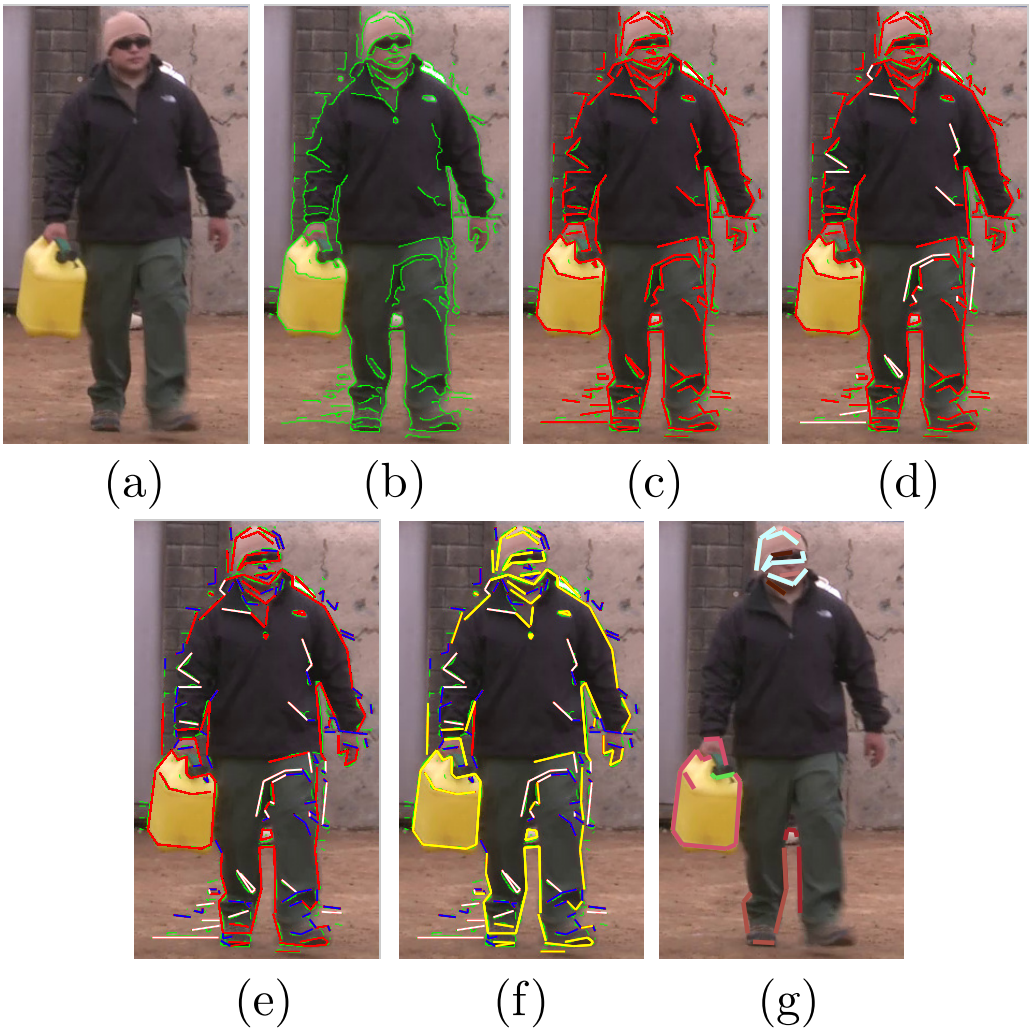


Figure 3.12: Illustration of non-person based pre-processing steps that reduce the number of edge lines (person-based are illustrated in Figure 3.11). Given an image (a) we obtain edges displayed with a green colour in (b). We then create edge lines that are displayed in (c) as red. White edge lines in (d) are as a result of applying the Colour Contrast approach to filter out edge lines that have a similar colour profile on both sides of their lines. Any short length edge lines are also removed presented as blue lines in (e). We then use any remaining red edge lines in (e) to perform level-wise mining in (f) where the yellow lines represents an edge line being a member of at least one edge chain. Any red edges in (f) means that they never became a member of a set in the level-wise mining process as they could not be merged with any other edge line to form a chain. Finally in (g) we present the edge chain boundaries created by the level-wise mining approach where some may be overlapping others. Note that if other preprocessing approaches related to the person, illustrated in Figure 3.11, were also used, additional edge lines would have been removed including the ones on the head.

Figure 3.13: Example output boundaries of our carried object detector across different datasets. Each boundary is displayed as a connected chain in red or blue for a better contrast with respect to the object colour or background.

3.8 Conclusion

In this chapter we present our novel approach to detecting carried objects. Our geometric carried object detector overcomes many limitations of other state-of-the-art detectors, outlined in section 2.1, such as not being limited to protrusion regions, not heavily relying on person models and most importantly being able to localise the boundary of the object accurately. It obtains these object boundaries by using edges in the scene and uses a level-wise mining heuristic to provide chains of edges. These edge chains define the object polygon which defines the object region. We then use this object region to represent the corresponding object detection by fitting a minimum enclosing rectangle to the object polygon. We also described various methods of reducing false positives in our detector and making it more efficient.

In the next chapter we describe how as a result of applying our detector, we use the obtained detections to locally connect them by incorporating spatial consistency.

Chapter 4

Tracking through Spatial Consistency

4.1 Introduction

In this chapter we present our Spatial Consistency Tracker (SCT). The goal of this tracker is to obtain a set of tracklets $\mathcal{T}$, maximising the number of true positive tracklets and minimising the false positive ones by means of spatial consistency. In addition to our target object, we also assume that there are entities in the scene that interact with our target object (typically a person or an aspect of the scene). We refer to these objects as *reference entities* and assume they have already been tracked. Therefore to achieve the above goal, the SCT tracker takes advantage of relationships based on spatial consistency between the target object tracklets and reference entities in $\mathcal{R}$. Encoding relationships at this early stage can remove a large number of false positive tracklets from $\mathcal{T}$, which will set the space of forming the object track hypothesis ω in Chapter 5. This becomes especially important for carried objects as they can vary dramatically in size, shape and colour, leading to a significant rise in false positives in addition to weak and partial detections due to high levels of occlusion. This makes tracking systems prone to false tracks and heavy fragmentation, as evidenced by applying state-of-the-art trackers to these detections.

To better capture true positive tracklets, we use spatially consistent events (e.g. carry, static), relative to an entity, to enforce a strong spatial prior distribution (represented as a *heatmap*) that encodes spatial consistency between an object and a reference entity. It must be noted that only spatial consistency can capture true positive tracklets, as only they follow

a consistent behaviour relative to their interacting entities during a spatially consistent event. This is not true however for spatially inconsistent events (e.g. drop, pickup), as in addition to true positives during these events, false positives also follow an inconsistent behaviour.

Therefore the SCT tracker mainly targets tracklets that undergo a spatially consistent event, however, it is not limited to such tracklets and merely gives more attention to them and can construct any tracklet that may be during any event. In the next section we present our SCT tracker and describe this attention driven mechanism.

4.2 Formulation

Other than the target object track that we would like to search for and find, there are other objects or entities in the scene that interact with our target moving object, referred to as *reference entities*. We assume all reference entities have already been tracked, where the reference entity tracks, or simply reference tracks, are provided in a given set of tracks $\mathcal{R}$. Here, an individual reference track $r \in \mathcal{R}$ is a time series of bounding boxes represented as $r = \{..., r^t, ...\}$. Each r^t represents a reference entity detection for a certain time t.

In this work, we make the simplifying assumption that spatially consistent events are the only events that govern the relationship between a reference entity r and an associated set of target object tracklets $\mathcal{T}$. That is, if an object tracklet $\tau \in \mathcal{T}$, where $\tau = \{d^a, ..., d^t, ..., d^z\}$ (a and z define the start and end frames of the tracklet respectively and t is a particular time frame during this interval), is associated with a reference track r, then there exists a bijective relationship between the corresponding detections $d^t \in \tau$ and $r^t \in r$. We also assume that the object tracklets are independent of each other.

Given a set of detections $\mathcal{L}$ from a detector, the process of which was described in Chapter 3, in addition to the set of tracklets $\mathcal{T}$, we denote a set $\mathcal{V}_\mathcal{T} = \{d \in \mathcal{L} | \forall \tau \in \mathcal{T}, d \notin \tau\}$ which consists of all other detections in $\mathcal{L}$ that are not a member of a tracklet. In an ideal case, the set $\mathcal{V}_\mathcal{T}$ will only contain true negative detections. To compute the set $\mathcal{V}_\mathcal{T}$, we assign it any remaining detections from subtracting all detections in the subsets in $\mathcal{T}$ from $\mathcal{L}$.

Our task is to find a set of object tracklets $\mathcal{T}$ that are associated with reference tracks in $\mathcal{R}$. Accordingly, we formulate our task as finding a subset of object tracklets $\mathcal{T}^*_\mathcal{R}$ from the set of all possible tracklets $\mathbb{T}$ that maximises the following objective function:

$$\mathcal{T}^*_{\mathcal{R}} \;=\; \arg\max_{\mathcal{T}\subseteq\mathbb{T}} \prod_{d\in\mathcal{V}_{\mathcal{T}}} (1\text{-}\mathcal{C}(d)) \prod_{\tau\in\mathcal{T}} \Big[\mathcal{C}_r(\tau,\mathcal{R};\theta_{\mathrm{c}})\mathcal{C}_s(\tau;\Theta_{\mathrm{s}})\Big] \tag{4.1}$$

In the above equation, $(1\text{-}\mathcal{C}(d))$ calculates the boundary cost of all detections $d \in \mathcal{V}_{\mathcal{T}}$, i.e. detections that are not part of an object tracklet. This boundary cost is obtained using the same cost function in Equation 3.1, described in the previous chapter in Section 3.3. By using $(1\text{-}\mathcal{C}(d))$ the objective function prefers to keep detections with low boundary costs (< 0.5) in $\mathcal{V}_{\mathcal{T}}$ and prevent them from becoming a member of a tracklet τ. This is further described in section 4.3.

The second cost $\mathcal{C}_r(\tau,\mathcal{R};\theta_{\mathrm{c}})$ captures the object-entity relationships between tracklets $\tau \in \mathcal{T}$ and reference entity tracks in $\mathcal{R}$ that are characteristic of certain events based on a spatial consistency model θ_{c}. This cost will be further described in section 4.2.1.

The third cost $\mathcal{C}_s(\tau;\Theta_{\mathrm{s}})$ parametrised by a smoothness model Θ_{s} regards a tracklet τ being more likely if the sequence of object detections constituting this tracklet are smooth with respect to motion and appearance. These measures are further described in section 4.2.2.

4.2.1 Object-Entity Relationship

A novel way of characterising objects given that certain events occur as a result of entities interacting with them, is that they follow an entity's trajectory, with a temporally continuous and characteristically consistent spatial relationship with respect to that entity. As a result, we regard a candidate object tracklet $\tau \in \mathcal{T}$ as more likely to be an object associated with a reference entity $r \in \mathcal{R}$, if the tracklet τ follows the trajectory of r with a spatially consistent behaviour that is characteristic of certain events.

It must be noted that in our SCT tracker, for any time t, each detection d^t is only in relation with one reference entity detection r^t. This knowledge of which detection is in relation to which entity is provided by our detector, Chapter 3, since the detector runs and provides detections for each entity in the scene, therefore knowing which entity each detection came from and is in relation to. We therefore capture this object-entity relationship in the second cost $\mathcal{C}_r(\tau,\mathcal{R};\theta_{\mathrm{c}})$ of Equation 4.1 which is expanded as follows:

$$\mathcal{C}_r(\tau,\mathcal{R};\theta_{\mathrm{c}}) = \prod_{d^t\in\tau} \mathcal{C}(d^t)^{(1-\mathcal{C}_h(d^t,r^t;\theta_{\mathrm{c}}))} \tag{4.2}$$

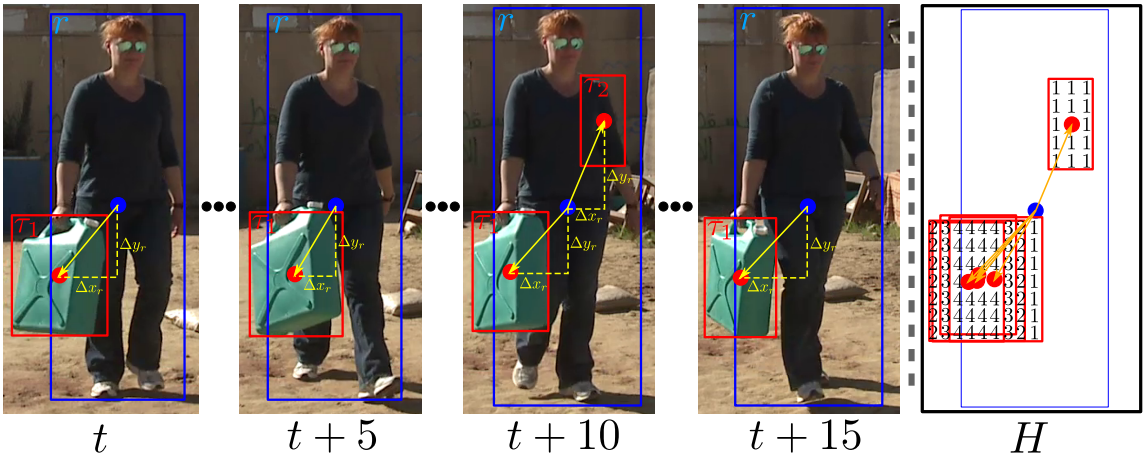


Figure 4.1: The process of building a heatmap H with tracklets τ_1 and τ_2 (red bounding boxes), relative to a person track r (blue bounding box). The relative location of each object detection is represent by an arrow which is drawn from the centre of the person (blue circle) to the centre of the object detection (red circle). All relative locations are then used in the heatmap H where each detection gives a vote to each offset pixel relative to a person. The total value of votes are indicated by a number in heatmap H. We can observe that due to spatial consistency between the person and the object, true positive object detections give rise to more votes while sporadic false positives do not.

Similar to the first term, $\mathcal{C}(d^t)$ is the boundary cost of a detection $d^t \in \tau$. The cost $\mathcal{C}_h(d^t, r^t; \theta_c)$ calculates a measure for an object τ to be spatially consistent with an entity r, where each of their corresponding detections, at frame t, must be spatially consistent. This measure of consistency is captured through a spatial consistency model θ_c. To quantify this model, we propose a voting measure that counts the number of times the relative position of a pixel with respect to the centroid of an entity's detection falls within an object detection.

As illustrated in Figure 4.1, let $\Delta x_{r^t}, \Delta y_{r^t}$ be the offset of a pixel relative to the centroid (x_{r^t}, y_{r^t}) of the entity's bounding box r^t at time t, i.e. $(x_{r^t} + \Delta x_{r^t}, y_{r^t} + \Delta y_{r^t})$ is the absolute position of the pixel relative to the image frame I^t. We define a function $\delta(\Delta x_{r^t}, \Delta y_{r^t}, d^t)$ as follows:

$$\delta(\Delta x_{r^t}, \Delta y_{r^t}, d^t) = \begin{cases} 1, & \text{if } (x_{r^t} + \Delta x_{r^t}, y_{r^t} + \Delta y_{r^t}) \in d^t \\ 0, & \text{if } (x_{r^t} + \Delta x_{r^t}, y_{r^t} + \Delta y_{r^t}) \notin d^t \end{cases} \tag{4.3}$$

The function $\delta(\Delta x_{r^t}, \Delta y_{r^t}, d^t)$ outputs a 1 or a 0 depending on whether the relative offset $(\Delta x_{r^t}, \Delta y_{r^t})$ is a member of detection d^t or not respectively. By using the above definition, we define a spatial consistency map H for each r, where the value of H at each relative offset position $(\Delta x_{r^t}, \Delta y_{r^t})$ is obtained by:

$$H(\Delta x_{r^t}, \Delta y_{r^t}) = \sigma(\frac{\sum_{\tau \in \mathcal{T}} \sum_{d^t \in \tau} \delta(\Delta x_{r^t}, \Delta y_{r^t}, d^t)}{|r|}, \theta_{\text{c}}) \tag{4.4}$$

Given a set of tracklets $\mathcal{T}$ associated with a set of entity tracks $\mathcal{R}$, the intensity values in H measure the number of votes for each relative offset pixel $(\Delta x_{r^t}, \Delta y_{r^t})$ given by the tracklets in $\mathcal{T}$. Since we expect objects to have a consistent relative location with respect to the entities (due to the nature of the specified events) and noise to be more randomly distributed, the spatial consistency map captures the locations relative to the entities where objects are most likely to exist. This is as a result of these locations receiving higher votes in the heatmap due to the repeated presence of potential carried objects even though they may be sparsely detected in the video.

After all votes for each offset position $(\Delta x_{r^t}, \Delta y_{r^t})$ is calculated, we obtain a value for the total number of votes. We normalise this value by the length of the corresponding relative entity, i.e. the duration of its frames, resulting in a ratio. A value is then estimated for each offset position ratio using a generalised logistic function σ which is calculated based on Equation 4.5 with an input x and a spatial consistency model θ_{c} defining its parameters. Any θ model for the generalised logistic function has parameters $\theta = \{A, B, C, K, M, Q, v\}$. The advantage of using the heatmap to model the object-person relationship can be observed in Figures 4.1, 4.2 and 4.3.

$$\sigma(x, \theta) = A + \frac{K - A}{(C + Q * e^{-B(x-M))})^{1/v}} \tag{4.5}$$

As a result of using Equation 4.4, we obtain a spatial consistency map H where each offset position relative to the centre of H (which corresponds to the centre of an entity's bounding box) has a value between 0.01 and 0.99 representing the likelihood of a pixel belonging to an object region, relative to an entity. This spatial consistency map can be visualised as a heatmap which is illustrated in Figure 4.2. The warm region represents the most likely location where an object may be, relative to the entity interacting with it. This heatmap distribution tends to get closer to the true distribution of the objects relative location with respect to an entity (e.g. for the object and person in Figure 4.2) as more tracklets are built, as further described in section 4.3.

We therefore regard a detection d^t as more likely to be an object, if the objects bounding box covers pixels with high intensity values in the heatmap. We model the *heatmap cost* $\mathcal{C}_h(d^t, r^t; \theta_{\text{c}})$ of equation 4.2 as:

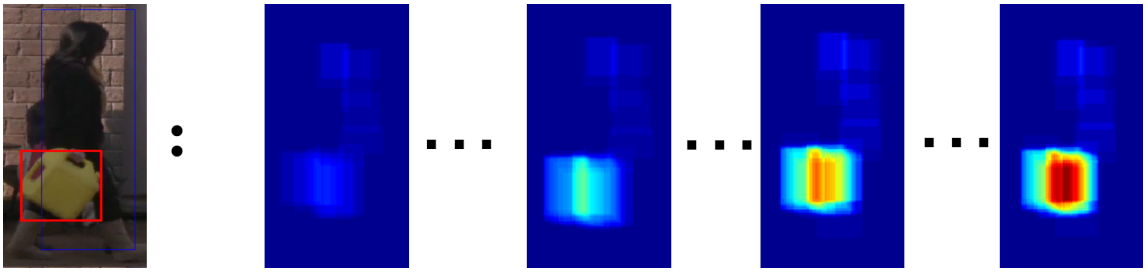


Figure 4.2: An illustration of the spatial distribution of a heatmap for a person being learned as the optimisation progresses. As more tracklets are built by the optimisation, the more accurate the heatmap becomes in its distribution. This distribution approximates the true relative position of an object to the person, where they can be seen in the leftmost image.

$$\mathcal{C}_h(d^t, r^t; \theta_c) = \frac{\sum_{(x,y)\in d^t} H(x\text{-}x_{r^t}, y\text{-}y_{r^t})}{|d^t|} \tag{4.6}$$

Therefore, to calculate the heatmap cost of a detection $d^t \in \tau$ based on the above equation, for each pixel (x,y) in the bounding box of d^t, we obtain the pixel's corresponding H value. We then take the average of all pixel H values by dividing by the total number of pixels, i.e. $|d^t|$. We use the resulting average value as an estimate of the heatmap cost for a detection d^t.

Due to the depth of the scene or inaccuracies of person detections, the track of an entity, e.g. a person, may vary in height or width over time. By obtaining the heatmap using these varying sized person detections, the relative location of the object to the person is not always consistent and the offset may be misaligned for other offsets over time. Therefore to obtain an accurate heatmap, we normalise and map each bounding box of a person track to a fixed size, while at the same time apply the same mapping to the object bounding box to preserve the object-entity relationship.

To perform this normalisation, as illustrated in Figure 4.3, we obtain a fixed sized bounding box to map all person detections to. We obtain this bounding box by taking the average width and height of all detections in a person track, represented by r^μ_w and r^μ_h respectively. This average bounding box is illustrated as green in Figure 4.3. For each bounding box in a person track r^t (blue bounding box) with a width and height of r^t_w and r^t_h, we obtain a width and height ratio of $\alpha = r^t_w/r^\mu_w$ and $\beta = r^t_h/r^\mu_h$ respectively. Based on the object-person relative offset Δx_{r^t} and Δy_{r^t}, we calculate a new offset for the object relative to the average bounding box as $\Delta x_{r^\mu} = \alpha * \Delta x_{r^t}$ and $\Delta y_{r^\mu} = \beta * \Delta y_{r^t}$. Since the width and height, d^t_w and d^t_h of the bounding box of the object also changes due to this mapping, we calculate the new width and height of the object as $d^\mu_w = d^t_w * \alpha$ and $d^\mu_h = d^t_h * \beta$. From

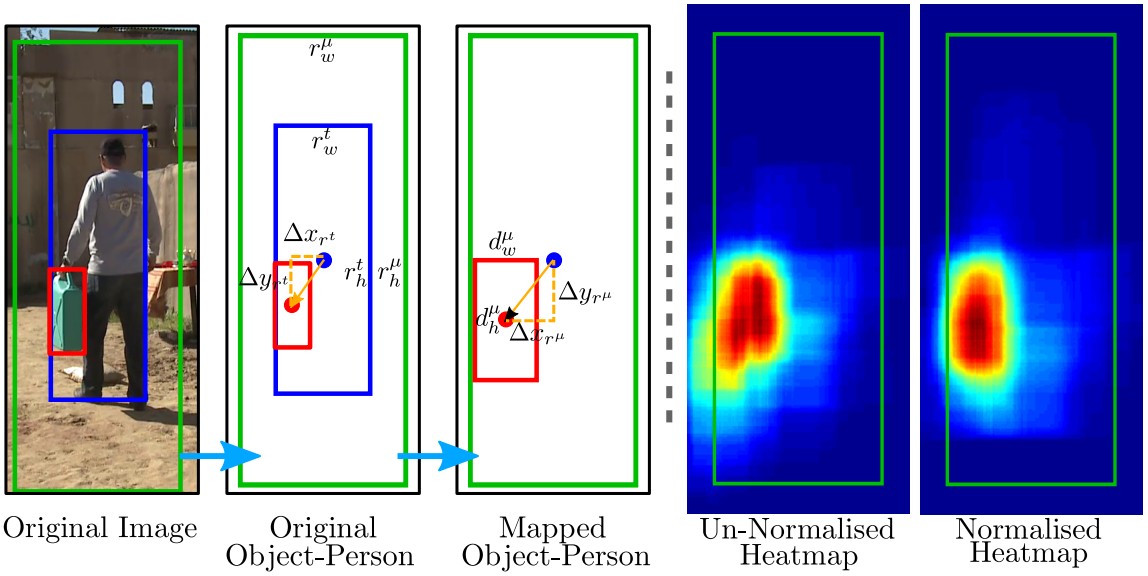


Figure 4.3: The process of mapping a person bounding box to a normalised person bounding box. The mapping is also applied to the object to preserve the object-person relationship. By comparing the normalised heatmap to the un-normalised heatmap, we can observe that the normalisation process greatly enhances the relative consistency as the normalised heatmap obtains a more localised distribution representing the true location of the object.

Figure 4.3 we can observe the result of this mapping where the mapped object-person relationship is preserved and is similar to the original object person relationship. Therefore, as a result of the aforementioned mapping, the normalised heatmap in Figure 4.3 obtains a more consistent and localised distribution, capturing the true location of the object relative to the person, while the un-normalised heatmap has a wider distribution.

The main goal of creating the heatmap and using the heatmap cost in the objective function is to *promote* true positive detections that have low detection costs into the tracking process. In addition to this, the heatmap also acts as an attention driven mechanism in the spatial consistency tracker. It does this by making the tracker focus more on true positive detections that are closer to the true location of the object (captured by the heatmap), while focusing less on false positive detections and subsequently mostly suppressing them. The role of the heatmap in SCT is described in greater detail in section 4.3.

4.2.2 Tracklet Suitability

The third term of equation 4.1, $\mathcal{C}_s(\tau;\Theta_s)$, calculates a cost based on the suitability of a tracklet, which is estimated in terms of three measures, namely (i) shape continuity, (ii) distance and (iii) path continuity. We therefore expand $\mathcal{C}_s(\tau;\Theta_s)$ in terms of these measures

as the following:

$$\mathcal{C}_s(\tau;\Theta_{\text{s}}) = \prod_{d^t,d^{t\text{-}1},d^{t\text{-}2}\in\tau} \mathcal{C}_{sc}(d^t,d^{t\text{-}1};\theta_{\text{sc}})\ \mathcal{C}_{dm}(d^t,d^{t\text{-}1};\theta_{\text{dm}})\ \mathcal{C}_{pc}(d^t,d^{t\text{-}1},d^{t\text{-}2};\theta_{\text{pc}}) \quad (4.7)$$

In order to build more suitable tracklets, we prefer connecting detections that are more similar in shape with regards to their bounding boxes. Therefore, the first term $\mathcal{C}_{sc}(d^t,d^{t\text{-}1};\theta_{\text{sc}})$ measures shape continuity. This measure calculates the shape similarity of the bounding box of a detection $d^t \in \tau$ against the bounding box of its previous detection $d^{t\text{-}1} \in \tau$. The amount of change is calculated by initially centring the two bounding boxes on top of each other and then calculating the following measure:

$$\mathcal{C}_{sc}(d^t,d^{t\text{-}1};\theta_{\text{sc}}) = \sigma(\frac{\text{Area}(d^t \cap d^{t\text{-}1})}{\text{Area}(d^t \cup d^{t\text{-}1})},\theta_{\text{sc}}) \quad (4.8)$$

In the above equation, the generalised logistic function σ with parameter values θ_{sc} estimates the shape continuity cost of two consecutive detections $d^t, d^{t\text{-}1} \in \tau$, based on the ratio of the area of their overlapping regions over the area of their union region. The higher this ratio is, the more similar their shapes are, leading to a higher cost.

Another suitable measure for building tracklets is the distance between consecutive detections in a tracklet. This measure should give higher cost to consecutive detections that are closer to each other, while penalising detections that are further away from each other with lower costs. Therefore, the second term $\mathcal{C}_{dm}(d^t,d^{t\text{-}1};\theta_{\text{dm}})$ calculates this distance measure based on the following equation:

$$\mathcal{C}_{dm}(d^t,d^{t\text{-}1};\theta_{\text{dm}}) = \sigma(D_2(d^t,d^{t\text{-}1}),\theta_{\text{dm}}) \quad (4.9)$$

In the above equation, we estimate the distance cost by using a generalised logistic function σ with parameter values θ_{dm}, based on the Euclidean distance between detections d^t and $d^{t\text{-}1}$, calculated by function D_2, illustrated in Figure 4.4 (a). If the distance is relatively short, a higher cost will be obtained.

Another important measure for a suitable track is that it is smooth and continuous with respect to its trajectory and path. As a result we calculate the path continuity cost $\mathcal{C}_{pc}(d^t,d^{t\text{-}1},d^{t\text{-}2};\theta_{\text{pc}})$ based on the following equation:

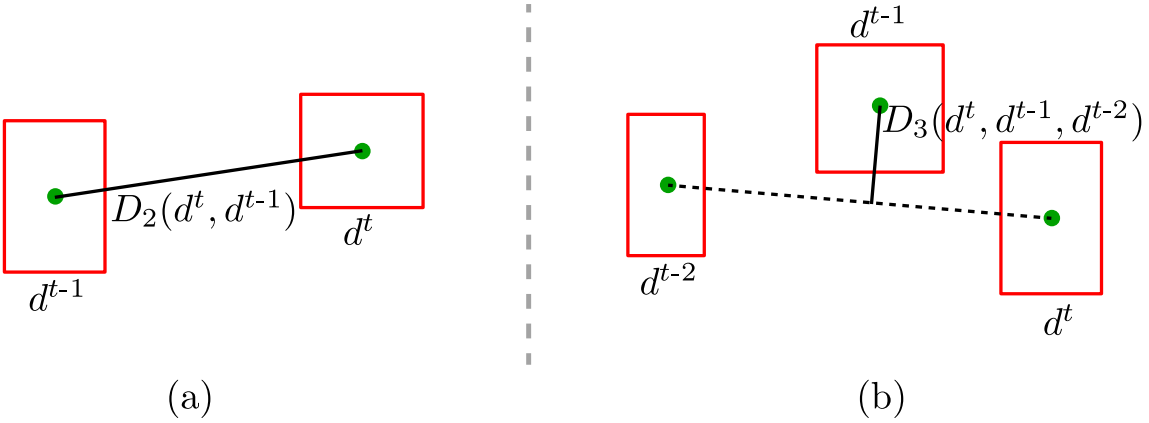


Figure 4.4: Examples of distance functions D_2 and D_3 used as part of the tracklet suitability measures. The distance is the length of the solid black line, obtained based on the centre points of detection d.

$$\mathcal{C}_{pc}(d^t, d^{t\text{-}1}, d^{t\text{-}2}; \theta_{\text{pc}}) = \sigma(D_3(d^t, d^{t\text{-}1}, d^{t\text{-}2}), \theta_{\text{pc}}) \tag{4.10}$$

The path continuity cost is estimated in the above equation using a generalised logistic function σ with parameter values θ_{pc}, based on the shortest distance between a point $d^{t\text{-}1}$ to the line between points d^t and $d^{t\text{-}2}$, calculated from function D_3, illustrated in Figure 4.4 (b). The shorter this distance is the smoother the path will be.

Using the objective function in Equation 4.1 which expands into the above aforementioned terms, we construct the optimal set of tracklets based on spatial consistency within an optimisation procedure, as described in the next section.

4.3 Spatial Consistency Optimisation

We now describe the optimisation procedure for the Spatial Consistency Tracker (SCT). The optimal solution of the optimisation problem with the objective function in equation 4.1, $\mathcal{T}^*_{\mathcal{R}}$, emerges as a result of iterations involving cyclic interactions between two main components of the objective function. We define the first component, $\mathcal{C}_s(\tau; \Theta_{\text{s}})$ in equation 4.1, as the cost dealing with the spatio-temporal continuity and suitability of a tracklet τ in the tracking process. The second component, $\mathcal{C}_r(\tau, \mathcal{R}; \theta_{\text{c}})$, is defined as the heatmap cost that models the object-entity spatial consistency relationship via the heatmap.

As illustrated in Figure 4.5, the cyclic nature between the aforementioned two components emerges during the SCT optimisation when longer tracklets are built from detections

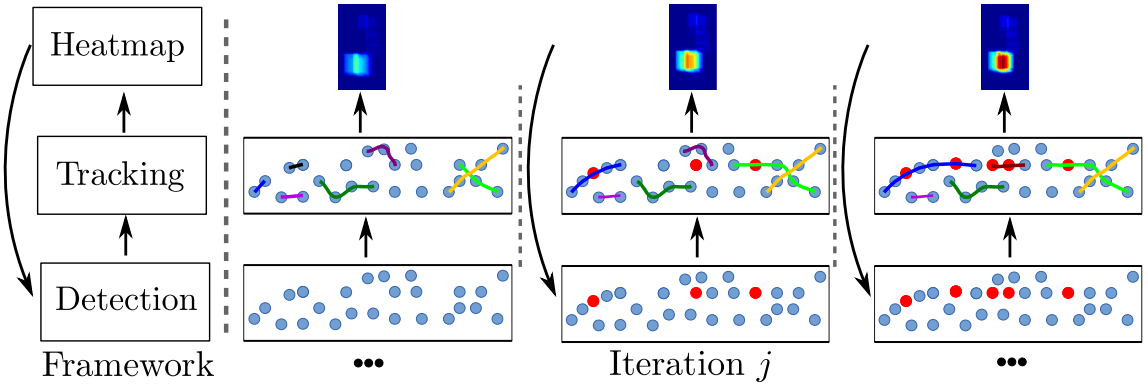


Figure 4.5: The architecture of our framework and examples of iterations in the optimisation process of our spatial consistency tracker as a result of applying the aforementioned framework. The cyclic nature between tracking and the heatmap emerges as more tracklets are built in the tracking stage resulting in a more accurate heatmap over more iterations, which in turn promotes more detections (red) for the tracking process.

in the first component, and as a result more detections from these tracklets will be available to contribute to the building of the heatmap in the second component, which in turn, the second component will promote more detections to be used in the first component. Therefore, due to this cyclic nature of our optimisation, longer and smoother tracklets of an object are built in each iteration, where subsequently each tracklet is also more spatially consistent to their interacting entity, while simultaneously contributing more tracklets to the creation of the heatmap, leading to more detections being promoted in the next iteration for creating longer and more spatially consistent tracklets.

Given a set of object detections $\mathcal{L}$, the SCT optimisation process is initialised with an initial empty hypothesis $\mathcal{T}^0 = \emptyset$ which subsequently leads to the set of unused detections $\mathcal{V}_\mathcal{T}$ being assigned all detections in $\mathcal{L}$, i.e. $\mathcal{V}_\mathcal{T} = \mathcal{L}$, since no detection is a member of a tracklet τ. Using $\mathcal{T}^0$ we obtain an initial cost based on the objective function in Equation 4.1. Since the objective function takes the product of a large number of values between zero and one, which result in a very small value, for implementation purposes we take the logarithm of the objective function which provides costs that are in a more suitable scale to work with.

The optimisation process starts with the initial hypothesis $\mathcal{T}^0$ and applies a sequence of *moves* to detections (whether they are part of a tracklet or not) to iteratively obtain a sequence of hypothesised tracklet sets $(\mathcal{T}^1, ..., \mathcal{T}^j, ...)$, based on a gradient descent based approach. Therefore, the objective function is used at each iteration j in the optimisation to decide whether to accept the new hypothesis $\mathcal{T}^j$ or to persist the previous hypothesis $\mathcal{T}^{j-1}$. In each iteration the move to be applied is chosen uniformly at random. The detection it is

applied to however is sampled based on a distribution calculated from the combined costs of detection and heatmap $\mathcal{C}(d^t)^{(1-\mathcal{C}_h(d^t,r^t;\theta_c))}$ in Equation 4.2, for each detection.

This distribution enforces the previously described attention mechanism in the tracking optimisation. Initially as the heatmap distribution is uniform, a detection is sampled primarily based on their corresponding detection costs. However, as the optimisation progresses the heatmap captures the true location of the object. The distribution obtained from the combination of the detection and the non-uniform heatmap cost gives more attention to detections that have a high heatmap cost. With this approach in sampling, a true positive with a low detection cost but a high heatmap cost may be sampled over a false positive detection with a high detection cost but a low heatmap cost, leading to the tracker's attention shifting from potential false positives to more true positives.

After a detection is sampled, we apply a move to the sampled detection and change the hypothesis $\mathcal{T}^j$ to create a new hypothesis $\mathcal{T}^{j+1}$. In total there are 6 moves, illustrated in Figure 4.6, which are namely (i) Birth, (ii) Death, (iii) Merge, (iv) Replace, (v) Split and (vi) Crossover. Each of these moves is described below:

(i) Birth: A sampled detection d from $\mathcal{V}_\mathcal{T}$ becomes a length one tracklet in $\mathcal{T}$. In Figure 4.6, first row and left to right, a green detection in $\mathcal{V}_\mathcal{T}$ becomes a tracklet τ_1 as a result of a *birth* move.

(ii) Death: A sampled detection d from any tracklet τ in $\mathcal{T}$ becomes a detection in $\mathcal{V}_\mathcal{T}$. If the tracklet is not of length one, the tracklet is broken into two tracklets. In Figure 4.6, first row and right to left, the tracklet τ_1 becomes a green detection in $\mathcal{V}_\mathcal{T}$ as a result of a *death* move.

(iii) Merge: A detection d from all subset tracklets in $\mathcal{T}$ is sampled. The tracklet $\tau = \{d^a, ..., d^t, ..., d^z\}$ of the corresponding sampled detection d^t is found. The first sampled detection is changed to the last detection in the tracklet, d^z. The detection d^z exists at frame z and based on this, a second detection is sampled from frame $z+1$. The second sampled detection's tracklet, τ' is found. Tracklets τ and τ' are merged and replaced as a new single tracklet $\tau'' = \tau \cup \tau'$ in $\mathcal{T}$. In Figure 4.6, second row and left to right, tracklet τ_1 is *merged* with tracklet τ_2, creating a longer tracklet τ_1 by assigning it all detections from τ_2.

(iv) Split: A detection d is sampled and its tracklet $\tau = \{d^a, ..., d^t, ..., d^z\}$ is found. τ is then split at the sample detection d^t and is replaced in $\mathcal{T}$ by the two new resulting tracklets $\tau_1 = \{d^a, ..., d^t\}$ and $\tau_2 = \{d^{t+1}, ..., d^z\}$. In Figure 4.6, second row and right to left, tracklet τ_1 is *split* into two shorter tracklets τ_1 and τ_2.

(v) Replace: A detection d_1 is sampled and its tracklet τ_1 is found, leading to a sampled detection d_1^t. A second detection d_2 is sampled from all detections in frame t and its tracklet τ_2 is found, leading to a sampled detection d_2^t. If d_2 was sampled from $\mathcal{V}_\mathcal{T}$ and isn't part of a tracklet, we assume it is a length one tracklet. d_1 then replaces d_2 in τ_2 and d_2 replaces d_1 in τ_1. In Figure 4.6, third row and in both directions, we can observe tracklet τ_2 swapping a single detection with tracklet τ_1.

(vi) Crossover: A detection d_1 is sampled and its tracklet τ_1 is found. A second detection d_2 is sampled from detections in existing tracklets that temporally overlap with τ_1. Similarly the tracklet of d_2, τ_2 is found. Based on the overlapping regions of τ_1 and τ_2, as illustrated in Figure 4.6, there are four cases to perform crossover on τ_1 and τ_2. The fourth case however has two potential ways of performing the crossover move and if it occurs, one of the two ways is chosen uniformly at random. The last case in Figure 4.6 provides examples of applying a *crossover* move based on the different ways two tracklets τ_1 and τ_2 can overlap.

The SCT optimisation generally starts by creating and accepting new length one tracklets as a result of applying the ***Birth*** move on detections that have detection costs higher than 0.5. This *start* in the optimisation is due to the way a detection cost is used in the objective function of Equation 4.1. For example, if a detection $d \in \mathcal{V}_\mathcal{T}$ has a cost of 0.8, since the objective function calculates 1-$\mathcal{C}(d)$ for any detection in $\mathcal{V}_\mathcal{T}$, detection d's contribution in calculating the objective function will be 0.2. However, if a ***Birth*** move creates a new tracklet $\tau \in \mathcal{T}$ out of detection d, the objective function calculates the likelihood of d as $\mathcal{C}(d)$ which results in a value of 0.8. Since by calculating the objective function, the 0.8 will lead to a higher overall hypothesis cost when compared to 0.2, the new hypothesis obtained by applying the ***Birth*** move will be accepted.

If the detection cost was 0.3 however, the reverse of this occurs since before applying a birth move, we will have a 0.7 while after we obtain a 0.3. Therefore the optimisation will reject the new hypothesis with a lower cost and continues to apply another move on the previous hypothesis. This may change however in later iterations of the optimisation when an accurate heatmap is obtained, where after applying a ***Birth*** move to the same detection that has a low detection likelihood of 0.3, but has a high heatmap cost of 0.8, the resulting hypothesis cost changes to $0.3^{(1-0.8)} = 0.78$ rather than the previous 0.3. Therefore this hypothesis with a detection that has a low detection cost will be accepted since the detection was promoted based on the heatmap.

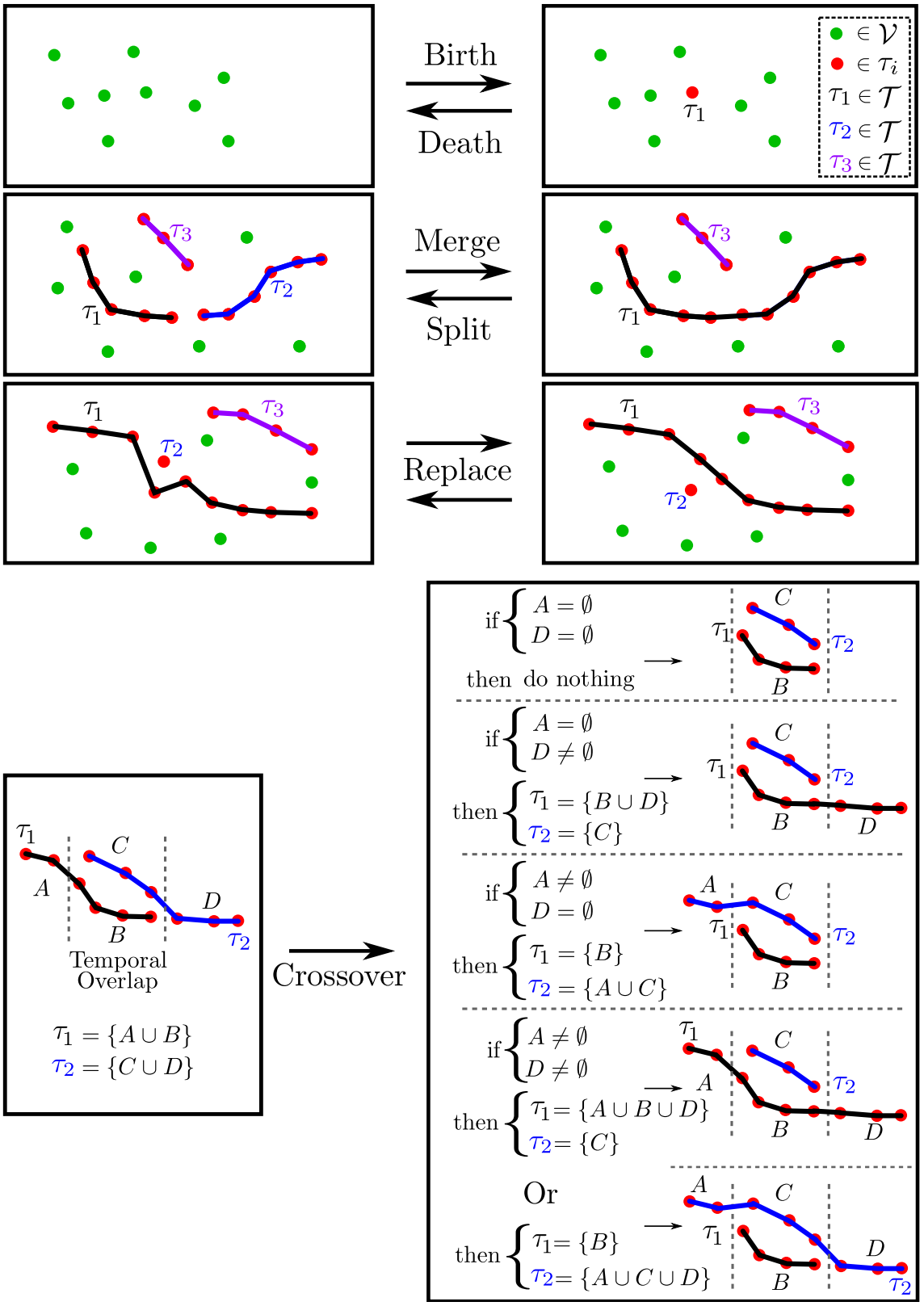


Figure 4.6: Examples of moves used in the SCT optimisation procedure. Each row showcases the affect of a move being applied to a set of unused detections in $\mathcal{V}_\mathcal{T}$ and a set of tracklets in $\mathcal{T}$.

Similarly, the optimisation applies other moves to iteratively create new hypotheses, where the cyclic nature between the two aforementioned components of the objective function leads to building longer tracklets that tend to better approximate true object tracklets. After a large number of iterations (8000), we terminate the optimisation process and regard the final set of tracklets of length more than one as the optimal set of objects tracklets $\mathcal{T}^*$.

4.4 Conclusion

In this chapter we present a novel approach to locally linking detections together that belong to a domain of generic object types which produce a larger number of false positives. The Spatial Consistency Tracker framework that incorporates this linking process uses context from the scene to suppress false positive detections while promoting true positives that were deemed weak by the detector. This framework primarily focuses on gathering and exploiting context that arises from the relationship between the object and the entity interacting with it. We model this relationship using spatial consistency in the heatmap which directly influences the detections as part of the tracklet building process. This results in a set of tracklets that represent the object with a more spatially consistent behaviour with respect to the entity interacting with it.

In the next chapter we present a framework which searches for the object trajectory using the obtained tracklets while incorporating event analysis jointly as part of the tracking process.

Chapter 5

Joint Tracking and Event Analysis

5.1 Introduction

When an object is being interacted with by an entity, it undergoes various changes in its trajectory depending on the nature of the interaction. These changes can be temporally categorised based on the motion of the trajectory, specially when considered relative to the interacting entity's trajectory. Using these relative relationships, one can describe how the object is interacted with in the scene at each frame. This description can be represented as a sequence of labels describing the interaction that occurs.

These labels provide the knowledge of which state an object is in at each frame. We define an event as a label that represents the state an object is going through. In other words, an event is an instance of a particular interaction between an object and an entity. Since each event describes the motion of the object trajectory and is unique in terms of its representation, one can use the knowledge of events as a type of context to aid in finding the trajectory of the object.

In this chapter we introduce our Joint Tracking and Event Analysis (JTEA) framework for tracking objects. The main novelty of JTEA is the improvement of object tracking by incorporating events to enforce spatio-temporal constraints on the tracking solution. Moreover, as a result of improved tracks, event recognition is also subsequently improved.

We formulate our JTEA approach under one objective function where tracking and event analysis are jointly performed, as described in the next section.

5.2 Formulation

We assume a tracker has produced a set of tracklets $\mathcal{T}$ that provide potential constituents for a single moving object within the target scene. Although there may be more than one, we are only interested in finding a single moving object. We again assume that there are *reference entities* $\mathcal{R}$ interacting with our target moving object that have already been tracked. The goal is to find the optimal object track consisting of a continuous sequence of tracklets, influenced by spatio-temporal relationships modelled by events between the target object and the reference object tracks in $\mathcal{R}$.

Each tracklet $\tau \in \mathcal{T}$ is a contiguous sequence of detections d, each represented by their minimal enclosing rectangle. A candidate track T_ω is defined by a subset of tracklets $\omega \subseteq \mathcal{T}$ such that there is no temporal overlap between tracklets in ω (we assume subsets are disjoint in what follows). The track T_ω is a time series of the detections that make up the tracklets of ω, linearly interpolated between the end of one tracklet and the start of the next if any frame gaps exist.

For a track T_ω we also obtain a corresponding sequence of event states defined as $S = \{s_1, s_2, ..., s_{||T_\omega||}\}$, where throughout this chapter $||.||$ denotes the number of elements in a given sequence or set. Each event state s takes an event type, defined in the set of event types $\mathcal{E} = \{1, 2, ..., E\}$, where E represents the total number of events we are interested in finding.

Our objective is to find an optimal set of tracklets $\omega^* \subseteq \mathcal{T}$ and an associated optimal sequence of event states S^* from the set of all possible event sequences $\mathcal{S}$, expressed as:

$$(\omega^*, S^*) = \underset{\omega \subseteq \mathcal{T}, S \in \mathcal{S}}{\operatorname{argmax}} P(\omega, S|\mathcal{R}, \mathcal{T}, \Theta_{\text{st}}) \tag{5.1}$$

$$= \underset{\omega \subseteq \mathcal{T}, S \in \mathcal{S}}{\operatorname{argmax}} P(\omega|S, \mathcal{R}, \mathcal{T}, \Theta_{\text{st}}) P(S|\mathcal{R}, \mathcal{T}) \tag{5.2}$$

In Equation 5.1 the term $P(\omega, S|\mathcal{R}, \mathcal{T}, \Theta_{\text{st}})$ evaluates the probability of each hypothesis set of tracklets ω and a sequence of event states S, given reference tracks $\mathcal{R}$, tracklet set $\mathcal{T}$ and a set of parameters Θ_{st}. The conditional probability for ω, $P(\omega|S, \mathcal{R}, \mathcal{T}, \Theta_{\text{st}})$, is defined in Equation 5.3 and is a product of three parts, namely *spatial*, *temporal* and *Gaussian observation*. The first two penalise non-smooth tracks and large gaps between tracklets, and the third is an event-state dependent Gaussian observation density over position and velocity. Although the spatial (smoothness) term may seem redundant alongside the Gaussian observation density, it is not localised to the current time instant by virtue of its construction with a smoothing function F, and we have found that it improves performance.

The employed smoothing function F is a moving average filter with a span of η .

$$P(\omega|S,\mathcal{R},\mathcal{T},\Theta_{\text{st}}) = \tag{5.3}$$
$$\Bigg(\underbrace{\prod_{i=1:||T_\omega||}\sigma(|T_\omega^i - F(T_\omega)^i|,\theta_{\text{s}})}_{\text{Spatial}}\Bigg)\Bigg(\underbrace{\sigma(\frac{\sum_{\tau_j\in\omega}||\tau_j||)}{||T_\omega||},\theta_{\text{t}})}_{\text{Temporal}}\Bigg)\Bigg(\underbrace{\prod_{i=1:||T_\omega||}\mathcal{N}(X_i|\mu^{s_i},\Sigma^{s_i})}_{\text{Gaussian observation}}\Bigg)$$

The *spatial* and *temporal* terms express the probability of a trajectory from ω, independent of the reference track $\mathcal{R}$ and capture standard tracking measures. The *spatial* term measures the degree of spatial association between temporally consecutive detections $T_\omega^i \in T_\omega$. It is calculated by taking the product of probabilities of a generalised logistic function σ with parameters θ_{s}, based on the absolute euclidean distances between each detection T_ω^i and $F(T_\omega)^i$. σ returns a value of one for shorter distances and decreases to zero for larger distances. F is a smoothing function applied to T_ω and $F(T_\omega)^i$ returns the smoothed corresponding point of T_ω^i. This term penalises the use of outlier tracklets.

The *temporal* term penalises the gaps between the tracklets that make up the track T_ω. We obtain this measure by applying a generalised logistic function σ with parameters θ_{t}, on the ratio of non-interpolated detections from tracklets in ω over the total length of the track T_ω. The larger the ratio, σ returns a value closer to one and if smaller, a value closer to zero. This measure promotes the use of observed tracklets over interpolated points.

The *Gaussian observation* term is where the events are taken into account. It enforces a prior distribution, based on bilateral relations between object track T_ω and reference tracks $\mathcal{R}$. We calculate these relations using a function $\mathcal{Q}$ with which we obtain an observation matrix X i.e. $X = \mathcal{Q}(T_\omega, \mathcal{R})$. We therefore calculate the Gaussian observation term as a product of the probabilities obtained from the normal distribution of individual observations X with respect to multiple event types $\mathcal{E}$, modelled by a mean μ^s and a covariance matrix Σ^s for an event s. The Gaussian observation term is further described in section 5.3.

For the second term in Equation 5.2, $P(S|\mathcal{R},\mathcal{T})$, we assume S is independent of $\mathcal{R}$ and $\mathcal{T}$ and define as a Markov chain on the state sequence. Thus, this term and the Gaussian observation term in Equation 5.3 effectively define a Hidden Markov Model (HMM), which is then coupled with the smoothness and gap-penalty terms to give the overall probability. The HMM captures the event analysis aspect of our framework, a method which is popular in event recognition work [76, 78, 98, 99, 94]. This HMM provides a measure of how likely pairs of an object track and reference tracks of possessive entities, conform to a model sequence of event states. Modelling the HMM is further described in the next section.

5.3 Modelling Events

We define our HMM model, illustrated in Figure 5.1, by a set of discrete events $\mathcal{E}$, an event variable $s_n \in \mathcal{E}$ at time n, transition probabilities between events $P_{u|q} = P_{s_n=u|s_{n-1}=q}$, $1 < u, q < |\mathcal{E}|$, prior probabilities for the initial event $P_{s_1=u}$ and output probabilities for each event $P_u(X) = P_{s_n=u}(X)$.

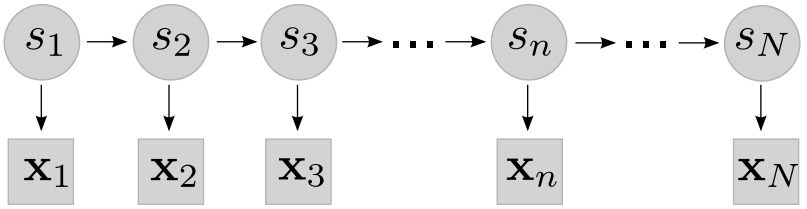


Figure 5.1: Our HMM model with observations X and event state variables s

The observation matrix X is composed of relative position and velocity relations, each captured in both horizontal and vertical directions, between the target object T_ω and the reference entity $\mathcal{R}$. Thus X has $4 \times ||\mathcal{R}||$ dimensions. These relative relations are calculated using function $\mathcal{Q}$ which takes the centre of the minimum enclosing rectangles of detections in both object and entity tracks as input.

We estimate the parameters of the HMM using maximum likelihood. For this we use a training dataset that is labelled with ground truth for the reference entity track, target object track and the events. Therefore to model each event type $e \in \mathcal{E}$, we obtain an observation matrix constructed by $\mathcal{Q}$ where only detections from the object and entity tracks are used that undergo the specific event e. Using these observations we model a Gaussian defined by a mean and a full covariance matrix μ^e and Σ^e respectively for each event state e. Modelled event Gaussians for two entities, namely person and scene, are illustrated in Figure 5.2 which capture object-scene (OS) and object-person (OP) relations for position (pos) and velocity (vel).

We also create a transition matrix based on the occurrence of an event u following an event q for all frames in the videos. Similarly we learn a prior for the occurrence of an event u. We therefore represent our HMM model as the set of Gaussians for each event, the transition matrix, and the prior.

Thus, to test a hypothesis track T_ω, given a set of reference tracks $\mathcal{R}$, we can construct a new observation matrix similar to above. By applying a Viterbi algorithm using this matrix and the above HMM model, we obtain an HMM measure for the *Gaussian observation* term along with a generated sequence of events for $P(S)$. We further describe the HMM model and the events used in Section 6.5.1.2.

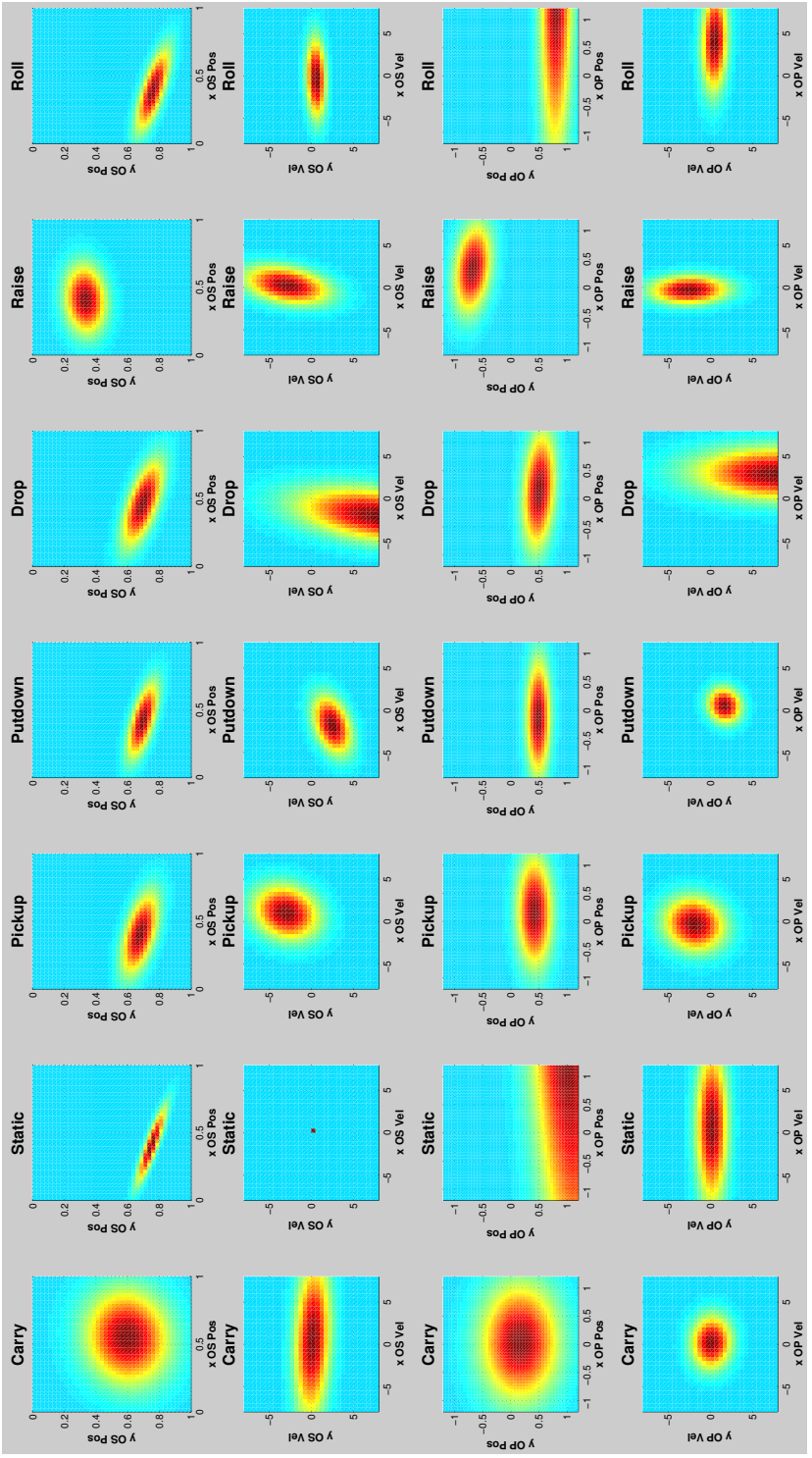


Figure 5.2: Gaussian observation densities in the HMM model for each event, based on the object-scene (OS) and object-person (OP) for position (pos) and velocity (vel) relations, in horizontal (x) and vertical (y) dimensions. For relative position, the image frame coordinates are set between zero and one, and for relative velocity the direction and magnitude of events is captured relative to the $(0, 0)$ coordinate, defining absolute consistency relative to the interacting entity. For example, in the bottom row, the object-person velocity relation for the *carry* event is highly consistent at $(0, 0)$, meaning the object follows the person consistently; while for the *raise* event is mostly horizontally consistent, however, it is vertically inconsistent in an upward direction.

5.4 Optimisation

The optimisation process is similar to Oh et al. [59] approach and the optimisation process outlined in the previous chapter in section 4.3, where we apply a set of *moves*, namely *add*, *remove* and *replace*, to construct successive track hypotheses. Given a set of object tracklets $\mathcal{T}$, obtained from any tracker, we initialise our object track T_ω by including only the first and last observed tracklets of $\mathcal{T}$ in our track hypothesis ω and obtain an initial probability using the objective function in Equation 5.2. Note that if these two tracklets are not suitable and do not belong to the optimal track hypothesis ω^*, they may be *removed* or *replaced* in the optimisation. In each iteration of the optimisation, a tracklet $\tau \in \mathcal{T}$ is randomly sampled, weighted by a normalised distribution of tracklet lengths.

A new hypothesis can be constructed in three ways depending on the sampled tracklet τ and the set of tracklets in ω: (i) if $\tau \in \omega$, we construct a new hypothesis by *removing* it from ω; (ii) if $\tau \notin \omega$ and it does not temporally overlap with any other tracklets in ω, we *add* it to ω and (iii) if it does temporally overlap, we *replace* any overlapping tracklets with τ in ω. Based on the *moves* above, at each iteration we construct a new track hypothesis T_ω. If the probability of T_ω (by Equation 5.2) is higher than the previous iteration's probability, we use the new hypothesis as the current best track hypothesis, if not, we continue with the previous best track hypothesis. Examples of two iterations in the JTEA optimisation process are illustrated in Figure 5.3.

By using this hill climbing approach, using a stopping criterion of a fixed number of iterations, the optimisation terminates and outputs the track hypothesis with the highest probability. Although this approach may only reach a local optimum, in practice we have found that the trajectories are suitable to represent the path of the object, as evidenced by our experiments in Chapter 6.

Events play a significant role in the Joint Tracking and Event Analysis (JTEA) optimisation as they are constructed from the track hypothesis in each iteration using the HMM, and they affect the suitability of new hypothesis tracks in future iterations through the objective function. This influence of events on the suitability of the track hypotheses primarily affects and is directed at the tracklets that are to be used in the hypothesis. As a result, the influence of the HMM emerges in the optimisation by choosing tracklets that enable the object track hypothesis to have a higher conformity to a sequence of modelled events when compared to other hypotheses consisting of other tracklets.

Illustrated in Figure 5.3, we present two consecutive iterations within our JTEA optimisation. Given a set of tracklets $\mathcal{T}$, at an iteration n, a temporally-disjoint subset ω is obtained and a contiguous track T_ω is produced by linearly interpolating across any

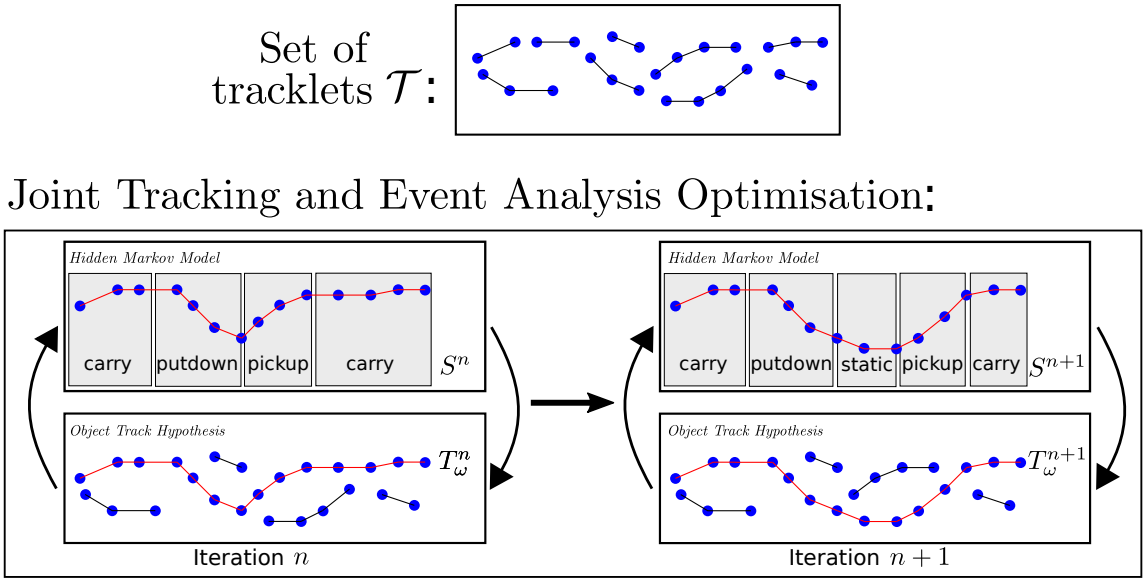


Figure 5.3: We illustrate two consecutive iterations within the Joint Tracking and Event Analysis optimisation. Given a set of tracklets $\mathcal{T}$, at each iteration, a temporally-disjoint subset ω is selected and a contiguous track T_ω is produced by linearly interpolating across any gaps. The Viterbi path S of event labels in the HMM is inferred from T_ω (arrow up), leading to an HMM measure (arrow down) and combined with the spatio-temporal factors in Equation 5.3 to give an overall probability. In the next iteration, a change to the subset ω is made and the overall probability re-computed. In this case, the new configuration is accepted since the probability is increased.

gaps between tracklets in ω. This hypothesis track is then given to the HMM (arrow up). The Viterbi path S of event labels in the HMM is inferred from T_ω and reference tracks in $\mathcal{R}$, leading to an HMM measure (arrow down) which is then combined with the spatio-temporal factors in Equation 5.3 to give an overall probability for the objective function. In the example provided in Figure 5.3, at iteration n, due to the motion in the middle of the trajectory T_ω, the HMM predicts a *pickup* immediately following a *putdown*. As it is unlikely to immediately pick up an object after it has been put down, the events produced from the trajectory do not conform to the model within the HMM and as a result a low HMM measure is given to the objective function. In this example, to consider the next iteration, we assume that this trajectory hypothesis is accepted at iteration n.

In the next iteration, $n+1$, a change to the subset ω is made as a result of applying a *replace* move. This produces a new hypothesis trajectory and the overall probability re-computed. In this case, the new trajectory produces events that better confirm to the modelled events in the HMM (emergence of the *static* event), thus obtaining a better HMM measure. As a result the new configuration is accepted over the hypothesis in the previous

iteration, since its probability from the objective function is increased.

As more and more tracklets are used as a result of the influence of the HMM to create improved object tracks with a higher conformity to modelled events, the improved tracks in turn improve the event sequence predicted in the HMM, subsequently improving the event analysis performed by the HMM. Thus through a joint optimisation tracking and event analysis influence and improve each other.

5.5 Conclusion

In this chapter we presented a framework for jointly performing tracking and event analysis where they mutually influence and improve each other. Given a set of tracklets, this framework aims at finding the most optimal object trajectory by taking into account the events that it produces. These events are outputted using a Hidden Markov Model along with an event measure which captures the conformity of the behaviour between the object track and an entity track, relative to ideal modelled interactions.

In the presented framework both tracking and event analysis terms are used within a single objective function as part of an optimisation where the tracks and events are iteratively improved. The improvement of tracks is primarily due to the influence of their events and the improvement of events are due to improved tracks.

In the next chapter we perform a quantitative and qualitative evaluation of the three frameworks outline in Chapters 3, 4 and this chapter which describe our Geometric Carried Object Detector, our Spatial Consistency Tracker and our Joint Tracking and Event Analysis frameworks respectively.

Chapter 6

Evaluation

6.1 Introduction

In this chapter we perform evaluations on each of the three main approaches presented in chapters 3, 4 and 5, namely our carried object detector, spatial consistency tracker and our joint tracking and event analysis framework respectively.

To accomplish this we use three datasets described in Section 6.2. We then describe our experimental setup in Section 6.3. We then provide qualitative and quantitative evaluations on each of the aforementioned chapters in sections 6.4 and 6.5.

6.2 Datasets

In order to evaluate the frameworks and approaches outlined in previous chapters, we use three datasets, namely PETS2006, MINDSEYE2012 and MINDSEYE2015. The PETS2006 dataset was chosen as a benchmark dataset and used for a baseline comparison to other state-of-the art approaches. A more complex dataset was also required leading to the use of the MINDSEYE2012 dataset which allows for a more in depth set of experiments. In order to evaluate both tracking and event analysis aspects of our joint tracking and event analysis framework proposed in Chapter 5, we created the new MINDSEYE2015 dataset. Each of these datasets is described in more detail in the following sections.

Figure 6.1: Sample images from the PETS2006 dataset. Target carried objects are indicated with a yellow rectangle.

6.2.1 PETS2006

The PETS2006 benchmark dataset was originally created with the goal of detecting left luggage in a train station scenario. This dataset consists of seven sets of videos, each set consisting of videos captured from four different cameras. The use of this dataset was later extended by Damen and Hogg [23], henceforth DHD, to detect carried objects on or around people. In this work only the third camera was used, due to its viewpoint, leading to seven videos with an average of 3000 frames. Each video was captured at 25 frames per second with a resolution of 720x576 where ground truth for many of the carried objects are provided. This dataset has become the benchmark of carried object detection and we apply our work to this dataset to compare against other state of the art approaches. Figure 6.1 illustrates examples of the PETS2006 dataset where carried objects are indicated with a yellow bounding box.

The ground-plane homography estimation of PETS2006 was provided as part of the dataset, which was needed to run the baseline state-of-the-art carried object detector of DHD. In order to extend the evaluation, a much larger number of person tracks were obtained when compared to the available person tracks from the dataset. These person tracks were obtained by first applying background subtraction (in the same manner as described in Section 3.2) to obtain foreground segmentations. By fitting a minimal enclosing rectangle to each connected foreground segmentation, the segmentations where then treated as person detections. By applying the off-the-shelf tracker by Pirsiavash et al. [63], person tracks

were obtained.

Although the benchmark PETS2006 dataset used in the literature is suitable for evaluating the performance of carried object detectors, most objects within this dataset are of the same type, there is only a single viewpoint and since it is an indoor scene, it does not have the challenges of outdoor scenarios. We therefore perform a more in depth evaluation of our framework on the MINDSEYE2012 dataset, as described in the next section.

6.2.2 MINDSEYE2012

To better evaluate our carried object detector and our spatial consistency tracker against other state-of-the-art approaches, we use the challenging MINDSEYE2012 dataset. This Dataset consists of 70 outdoor video clips created by a third party from the Mind's Eye project Year 2 dataset [25], where each video consists of either a person carrying an object or walking through the scene without one. The complexity of this dataset results from variations in camera settings, environmental factors such as changes in lighting conditions (e.g. brightness due to weather), camera blur, shadows, moving trees, grass and cloths in the background, various person trajectories relative to the camera (e.g. walking in front of or towards the camera) and most importantly a much larger variety of carried object types. The videos were captured at 30 frames per second with a resolution of 1280x720 with an average length of 200 frames for each video.

The person tracks and ground truth carried object tracks were provided with the dataset while the ground-plane homography estimation required for running DHD was done for each camera setting. Figure 6.2 illustrates sample images from the MINDSEYE2012 dataset.

6.2.3 MINDSEYE2015

Carried object detection datasets, like PETS2006 and MINDSEYE2012, typically include only people walking with or without carried objects. Our joint tracking and event analysis framework however is designed and expected to perform when people interact with objects in a variety of ways as described in Chapter 5. For this reason, we created the MINDSEYE2015 dataset, by selecting a subset of videos from the Mind's Eye Year 2 dataset [25], where in each video, the carried object undergoes a variety of interactions performed by the entities in the scene. It must be noted that the subset of videos used in MINDSEYE2015 are different to the subset used in the MINDSEYE2012 and that they do not have any videos in common.

Figure 6.2: Sample images from the MINDSEYE2012 dataset. The top three rows of images illustrate frames of people with carried objects which are indicated with a yellow rectangle. The fourth row provides examples of people not carrying objects. These examples also illustrate the challenges of the dataset, e.g. shadows, background motion, lighting conditions and the different scenarios the videos are captured from.

MINDSEYE2015 consists of 15 videos (5 recordings captured from 3 different viewpoints), each lasting approximately 6000 frames. These videos were converted to a resolution of 640x360 at 20 frames per second. Excluding frames where no event occurs (empty scenes), there are approximately over 45 minutes of interactions between objects and entities. The videos are taken from three different viewpoints, illustrated in Figure 6.3, which allows a better evaluation of the capabilities of our approach in dealing with object occlusions. The viewpoints offer different types of challenges to a tracker: viewpoint C1 has medium levels of object occlusion and medium levels of scene depth; viewpoint C2 has medium levels of object occlusion (when the object is held in front of the person) and high levels of scene depth; viewpoint C3 has high levels of object occlusion (depending on which side of the person the object is carried) and low levels of scene depth.

Videos in the dataset show a variety of people interacting with various different objects. In the majority of frames there are at most one person and one object, but there are cases of more or less than one person or object present in the scene. It is also worth noting that, since the dataset was captured outdoors, the movement of trees and cloths on the table, as well as the change in brightness of the video due to clouds and distance of the person to the camera cause challenges for object detectors.

We have defined various events to allow for a full description of the scene with regards to the state of the carried object, from the start of its appearance to its disappearance. These events are further described in Section 6.5.1.2.

Ground truth for person tracks, carried object tracks and events are fully annotated. This dataset has been made publicly available [84] with all ground truth annotations. It also provides the carried object detections, tracklets, final tracks and automatic person tracks obtained from applying the frameworks outlined in previous chapters. Since the videos in this dataset are very long, we divide each video into clips defined by the start and end of each ground truth carried object track. Therefore each clip has a target carried object to be tracked.

6.3 Experimental Setup

In this section we describe the parameter settings and the evaluation measures used in evaluating the different frameworks presented in the previous chapters.

(a) C1 (b) C2 (c) C3

Figure 6.3: Sample images from the MINDSEYE2015 dataset. Each row illustrates an interaction between the same object and person at an exact moment in time from the three different viewpoints of C1, C2, and C3. The different viewpoints are not synchronized and the videos only temporally overlap for each row of viewpoints. Various types of occlusion, e.g. partial or full occlusion, can also be observed in the example images where the target object is localised by a yellow bounding box.

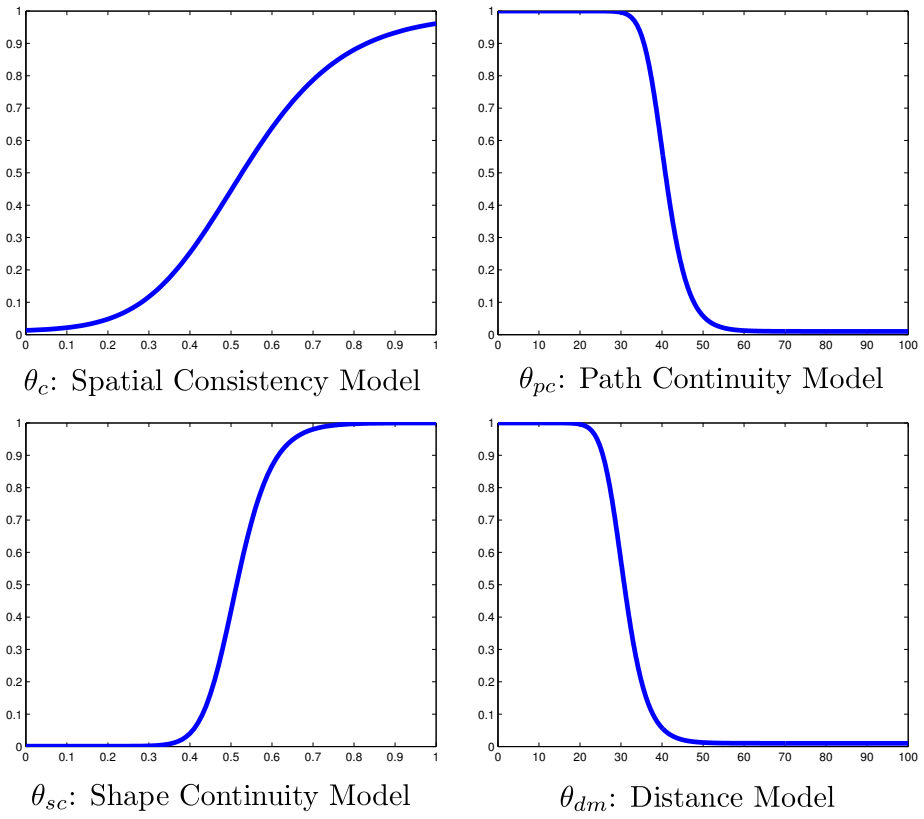


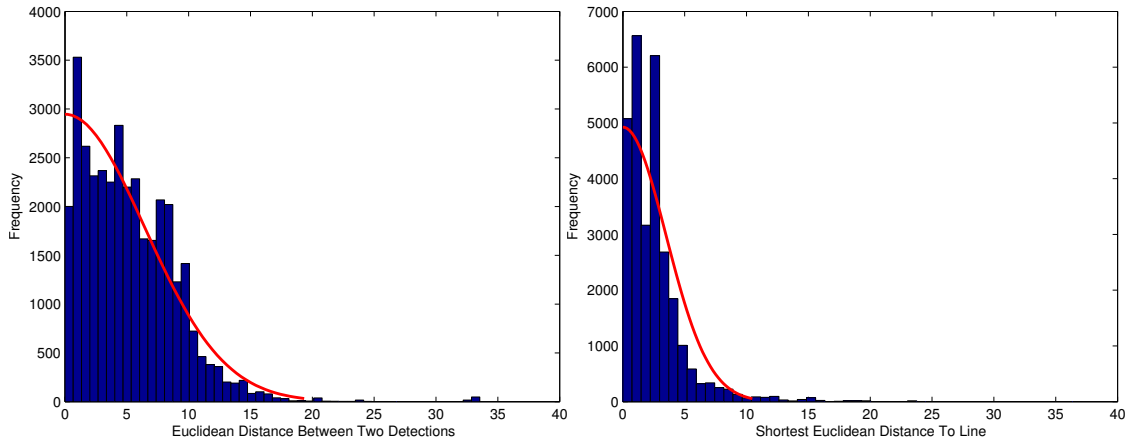
Figure 6.4: Distributions of different models used by the generic logistic function.

6.3.1 Parameter Settings

Throughout Chapters 4 and 5, various θ where employed as part of various objective functions, defining the model of the generic logistic function, in terms of its parameters in Equation 4.5. The primary role of these models and the generic logistic function was to normalise the costs of different terms in the objective functions to a value between zero and one, often incorporating a non-linear distribution. In Table 6.1 we define the employed models in this thesis by the generic logistic function by numerically presenting their parameter values. We also present each model's graphical representation illustrating its distribution in Figure 6.4.

θ	description	A	B	C	K	M	Q	v
θ_{c}	Spatial consistency model	0.01	7	1	0.99	0.5	0.5	0.5
θ_{pc}	Path continuity model	1	0.3	1	0.01	40	0.4	0.4
θ_{sc}	Shape continuity model	0.001	20	1	1	0.7	0.01	0.5
θ_{dm}	Distance model	1	0.3	1	0.01	30	0.4	0.4

Table 6.1: Various parameters of models used by the generic logistic function.



(a) (b)

Figure 6.5: Generic logistic function parameter learning for distance and path continuity models using ground truth object tracks.

The spatial consistency model θ_{c} follows a distribution that is approximately linear, however we penalise objects that are less convex, i.e. < 0.5. slightly heavier and promote objects that are more, i.e. > 0.5, as illustrated in Figure 6.4. For the shape continuity model θ_{sc}, we consider an overlap ratio value of 0.5 as the centre of the distribution where similar to before we penalise and promote other ranges in the distribution, albeit more heavily.

The parameters in the distance and path continuity models were based on learnt distributions from the ground truth, as illustrate in Figure 6.5. For the distance model, Figure 6.5 (a) displays a histogram of Euclidean distances between consecutive detections in ground truth object tracks. The red line illustrates the half-normal distribution fitting to the data. We can observe that the distribution covers distance values of up to 20. This indicates that longer distances are unlikely distances between consecutive true positive detections in an object track. As a result we set the generic logistic function parameters for the distance model θ_{dm}, Figure 6.4, such that distances less than 20 obtain a high (good) cost. After the value of 20 however, the distribution cost slowly drops until 40 so that we do not heavily penalise object tracks that may have imperfect true positive object detections, as they may naturally arise from the detector.

We obtain a similar histogram for the path continuity model where the distribution converges at a value of 10. However, since the ground truth tracks are highly smooth with respect to their trajectories, in practice we found that tracks obtained from imperfect detections are not as smooth. Therefore to take this into account, we increased the value of the distance of an ideal smooth trajectory from 10 to a value of 30 in the path continuity model in Figure 6.4. Simalar to the distance model, the distribution is gradually reduced as

the distance increases. It must be noted that values of zero were removed when creating the histograms as they would have been highly dominant in the distribution. All other θ models in this thesis follow a linear distribution.

In our experiments, the parameters presented in Table 6.1 and all other parameters and thresholds presented in this thesis were tuned on a subset of the Mind's eye Year 2 evaluation videos, non of which exist in the MINDSEYE2012 and MINDSEYE2015 datasets. Values of these parameters and thresholds are independent from any particular selection of subset. This is because general geometric properties such as the convexity model θ_{con} are invariant across samples from any dataset. Moreover, as the focus and use of our carried object detector is to prove a concept, that is to illustrate the benefits of employing a generic model as part of an object detector, only the convex shape model is investigated throughout the experiments in this chapter. We also set the span parameter for the smoothing function, described in Section 5.2, to $\eta = 5$.

In order to run any of the state-of-the-art approaches employed as part of experiments described in this chapter, default parameter settings of each approach were used as it is often considered most suitable for general uses.

6.3.2 Evaluation Measures

The evaluation measures used in this chapter concentrates on the tracking performance and is thus done with respect to the spatio-temporal localisation of object detections in object tracks, based on a frame by frame comparison to ground truth.

Therefore, to evaluate and measure the tracking performance of a certain approach based on the experiments outlined in this chapter, we primarily use the $F1$ score as a measure for comparison. We calculate the $F1$ score using Equation 6.1:

$$F1 = \frac{2 * \text{True Positive}}{2 * \text{True Positive} + \text{False Positive} + \text{False Negative}} \tag{6.1}$$

In the above equation, we define a detection d as being a true positive if its overlap measure with its corresponding frame ground truth detection d_{gt} is more than a certain threshold γ. We define this overlap measure as the following:

$$\text{Overlap Measure} = \frac{Area(d \cap d_{gt})}{Area(d \cup d_{gt})} \tag{6.2}$$

In the related work, Damen and Hogg [24] and Dondera et al. [28] have used 0.15 and 0.2 as values for the threshold γ respectively. That is, they consider a detection as a true positive if it has an overlap measure of more that 0.15 or 0.2. In our evaluation we report

tracking results on the full range of overlap thresholds, i.e. $0.01 < \gamma < 1$. This allows for a more accurate evaluation on the different detectors and the quality of the detections that they output.

In the following sections we present various experiments incorporating the aforementioned parameter settings and evaluation measures as part of evaluating different frameworks presented in this thesis.

6.4 Evaluation of Detection and Tracklet Building

In this section we perform experiments to evaluate the suitability of our geometric carried object detector and Spatial Consistency Tracker, based on the datasets outlined in Section 6.2 and the experimental settings outlined in Section 6.3.

These experiments consist of two aspects, first of which is a quantitative analysis in Section 6.4.1, where we evaluate our Spatial Consistency Tracker (SCT) against the state-of-the-art protrusion based Damen and Hogg [24] carried object detector (DHD). This quantitative evaluation is performed in terms of the tracklets produced by both approaches. To further illustrate the true potential of our SCT approach, we also evaluate against variations of our SCT approach consisting of alternate key components of this tracker and illustrate the benefits of incorporating spatial consistency within this framework.

Secondly, in Section 6.4.2, we perform a qualitative analysis of detection comparisons between our carried object detector described in Chapter 3 and the detections of DHD. Moreover, we also present an experiment comparing heatmaps as a result of applying different variation of the SCT tracker, highlighting the importance of the final SCT framework.

6.4.1 Quantitative Analysis

In this section we present a quantitative analysis of the performance of our spatial consistency tracker. The experimental setting of this analysis is described in the following section.

6.4.1.1 Experimental Settings

The main experiment is to evaluate the performance of our SCT approach against the state-of-the-art DHD approach. Additionally, we also evaluate against variations of our SCT approach where the architecture of each is illustrated in Figure 6.6. The experimental setting of each variation is described as the following:

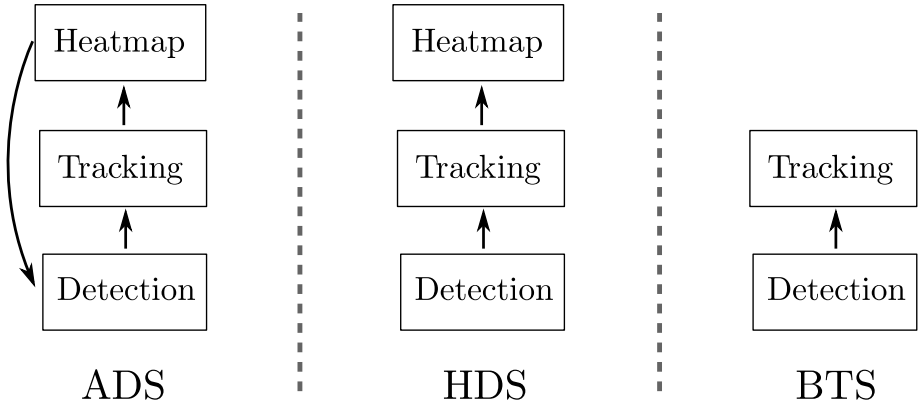
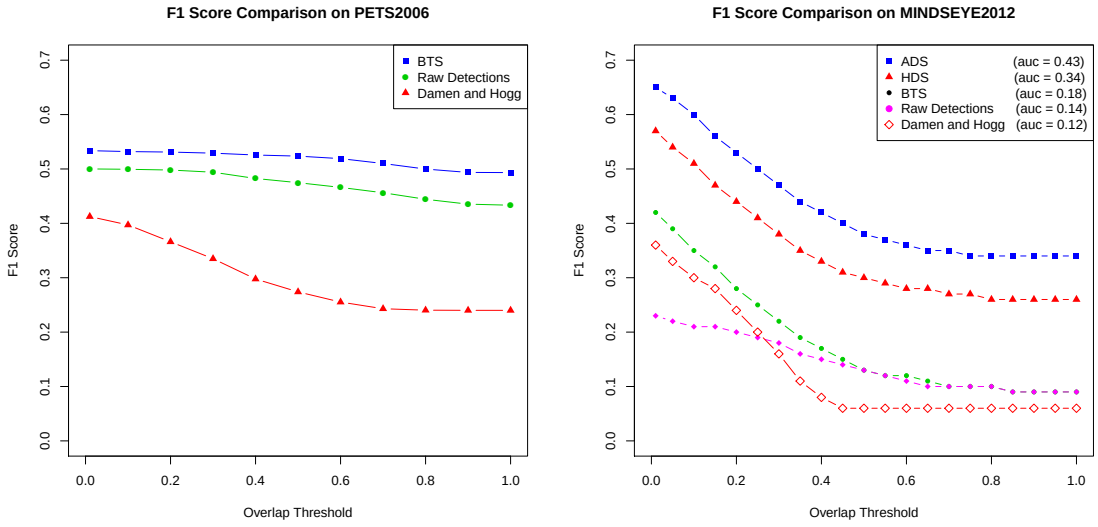


Figure 6.6: Different variations of the SCT approach for evaluation. Attention Driven System (ADS) represents the full SCT framework where the heatmap promotes True positives and gives more attention to them. The Heatmap Driven System (HDS) uses the heatmap to filter tracklets in a post-processing stage and does not allow the heatmap to influence the tracking optimisation in any way. The Basic Tracking System (BTS) is a basic tracker that does not use nor creates the heatmap.

- **Attention Driven System (ADS):** The ADS architecture captures the full framework presented in Chapter 4 describing our spatial consistency tracker. This framework capitalises on the potential of using the object-entity relative positional relationship, captured via the heatmap, to build tracklets. Additionally, the heatmap also introduces an *attention-like* mechanism into the optimisation process, where the heatmap shifts the sampling distribution of detections for the tracklet building process from all detections with high detection costs to detections that are more likely to be on the true location of the object, relative to the entity interacting with it. This attention-like mechanism effectively makes the optimisation apply more moves, as part of the tracklet building process, to detections that are more likely to be true positives.

- **Heatmap Driven System (HDS):** To highlight the important role of the heatmap in the ADS architecture and the benefits it provides, the HDS architecture removes the effects of the heatmap in the optimisation process on the detections (i.e. no arrow down in the architecture of HDS, in Figure 6.6) in two ways. Firstly it removes the benefits of using the heatmap and promoting detections at each iteration of the optimisation and the iterative nature of building the heatmap. Secondly, it avoids using the *attention-like* mechanism in the optimisation for suppressing the false positives. Therefore in the HDS architecture, the aforementioned two benefits of the heatmap are removed as there is no influence from the heatmap to the detections. The heatmap is therefore only used in a post-processing stage where it is built after the



(a) PETS2006 (b) MINDSEYE2012

Figure 6.7: Evaluation of our Spatial consistency tracker in terms of $F1$ scores as the threshold of overlap increases on both PETS2006 and MINDSEYE2012 datasets.

tracking process has been completed and only has the role of filtering out tracklets that are not on the true location of the object.

- **Basic Tracking System (BTS):** To evaluate the tracking aspect of the spatial consistency tracker independently and without the influence of the heatmap, the BTS architecture is used. This architecture represents a framework where the tracklet building process is completed during the optimisation without the creation of the heatmap or its use, effectively setting the heatmap cost of $\mathcal{C}_h(d^t, r^t; \theta_c) = 0$ in the objective function in Equation 4.2 in Section 4.2.1.

6.4.1.2 Results & Conclusions

The results illustrated in Figure 6.7a provides $F1$ curves for the performances of DHD, raw detections (RD) and BTS, applied on the PETS20016 dataset for a quantitative evaluation of the detections obtained from the aforementioned approaches. It can be clearly observed that our detector significantly outperforms the detections obtained from DHD, whether they are evaluated simply as detections, i.e. RD, or as part of object tracklets (BTS). It is also worth noting that for stronger overlap thresholds, i.e. closer to 1, our approach obtains more consistent true positives due a more constant $F1$ score when compared to the sudden drop observed in DHD. This indicates that by using our approach we obtain object

detections that more accurately localise the carried objects when compared to detections from DHD which produce larger boundaries for the objects. This can also be observed in Figure 6.8 images (a,d,k) as part of the qualitative analysis.

As well as DHD, RD and BTS, Figure 6.7b additionally shows results of HDS and ADS, for a quantitative evaluation of our spatial consistency tracker after applying it on the MINDSEYE2012 dataset. Here we can observe that our BTS approach again outperforms DHD, while RD only outperforms DHD after an overlap threshold of > 0.25. Once more this highlights the large object boundaries obtained from DHD which are initially counted as true positives for overlap thresholds of < 0.25. Overall, we can again conclude that our detections outperform the detections of DHD.

While BTS outperforms DHD, we can observe a significant tracking improvement by using the heatmap as part of our spatial consistency tracker, whether it is used in HDS or ADS. This highlights the importance of using the object-entity relation as part of the tracking process, captured via the heatmap. Additionally, by comparing the performance of ADS and HDS we can conclude on the importance and benefits of updating the heatmap and its influence on the tracklet building process, as a result of applying them in each iteration of the optimisation in ADS, when compared to the heatmap's creation as only a post-processing stage in HDS. Moreover, this also highlights the benefit of the promotion of detections via the heatmap cost and the attention-like mechanism the heatmap provides as part of the spatial consistency tracker optimisation.

6.4.2 Qualitative Analysis

In this section we present a qualitative analysis of the detections provided by our carried object detector and the ones provided by DHD. Additionally we perform a comparison on the quality of the heatmaps provided by each variation of the SCT tracker. The Experimental Settings of this analysis is present below.

6.4.2.1 Experimental Settings

As illustrated in Figure 6.8, we present the detection results of our carried object detector, described in Chapter 3, applied on the PETS2006 and MINDSEYE2012 datasets. We compare these results to detections obtained by applying DHD. We perform a qualitative analysis by summarising success and failure cases of both our detections (blue bounding box) and DHD (red bounding box), with respect to the ground truth (green bounding box).

In Figure 6.9 we present two heatmaps, the first was obtained during the ADS architecture where the heatmap updates and influences the tracking process at each iteration. The

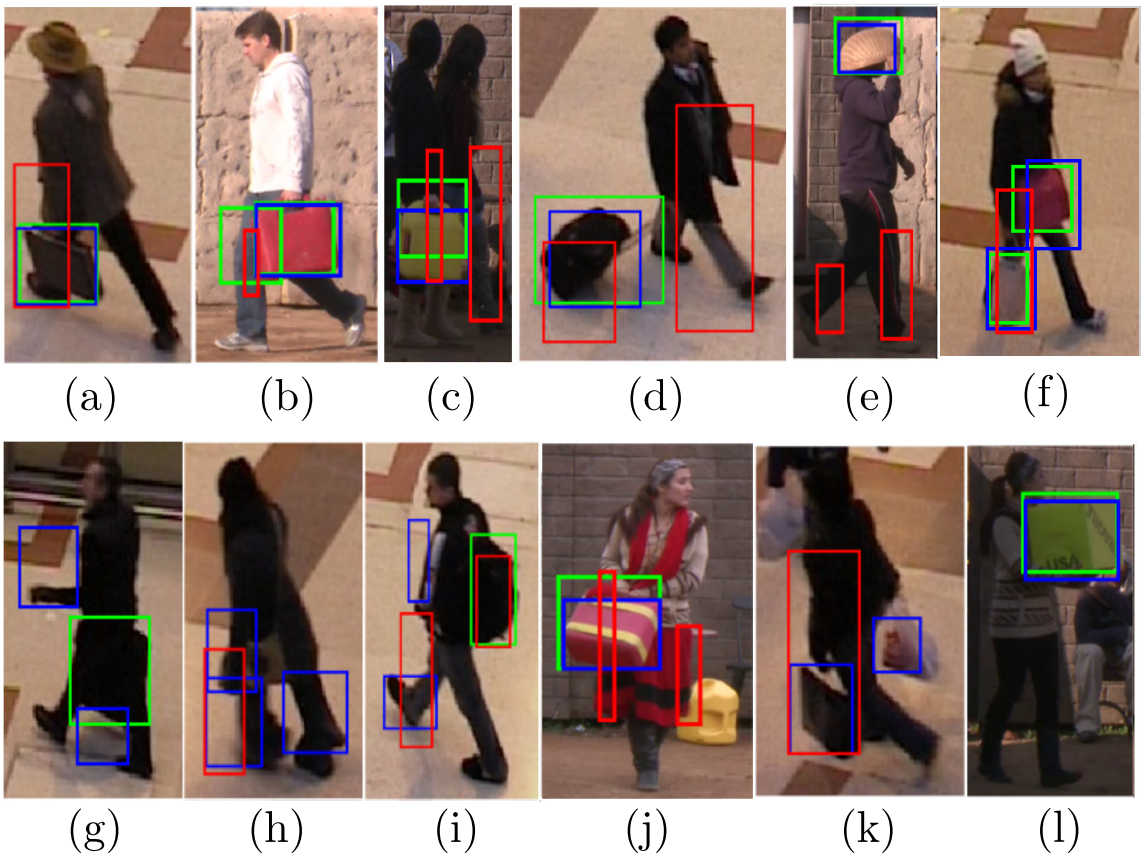


Figure 6.8: Illustration of the successes and the failures of our carried object detector and a comparison with a baseline state-of-the-art detector Damen and Hogg (DHD) [24]. For images (a)-(l), boxes coloured in green correspond to ground truth, red to baseline and blue to those obtained using our approach.

second heatmap is based on the HDS architecture where the tracking process is completed and the heatmap is created from the tracklets in a post processing stage. A quantitative analysis of the results from both of the aforementioned experiments is presented below.

6.4.2.2 Results & Conclusions

In Figure 6.8 images (a)-(f) and (j)-(l) illustrate how our approach is able to detect different types of objects such as boxes, bags, plastic bags and suitcases, while the baseline DHD approach is unable to. This highlights the merits of performing generic object detection without specific object models. (g)-(i) show cases where our approach performs poorly, as the edges do not sufficiently delineate the object from the person while also obtaining many false positives. (j) illustrates a case where edges on top of the object are not found and only a partial detection is obtained even though it is covering most of the ground truth

bounding box.

The (b,c,f,j) images illustrate that our approach is also able to detect objects that are not on protrusion regions. This is not the case however for the baseline Damen and Hogg detector as highlighted in images (a,b,c,e,j) which relies on protrusion and cannot detect objects that are on the person region. (c,d) illustrates situations where multiple people are in close proximity, or when the person's bounding box is displaced, heavily affecting DHD. (e) illustrates a case where the influence of a relatively strong prior on the position of the object in relation to the person can hinder the detection of an object (e.g. basket) above a person's head. Note that the object ground truth is sometimes not available for all people in the PETS2006 dataset e.g. (k).

So that others may use or build upon our carried object detector, we have made a basic version of our carried object detector and made it publicly available [83]. This version obtains edge lines from a sample image and performs level-wise mining to construct and provide object boundaries.

In order to qualitatively analyse the benefits of updating the heatmap iteratively within the SCT optimisation and to highlight the advantages of its attention-like mechanism, Figure 6.9 illustrates two heatmaps obtained from the ADS and the HDS architectures. As a reference to where the true location of the object is relative to the person, the left most image (*Detections*) shows the location of the object and the person. Additionally, object detections for the single image frame are also illustrated as red rectangles, where brighter and darker rectangles represent detections with higher and lower costs respectively.

By comparing the ADS and the HDS heatmaps in Figure 6.9, we can observe that the heatmap from the ADS approach suppresses the false positives on the person's upper body clothes from becoming a strong region in the heatmap, as a result of preventing the creation of their tracklets. This is due to the attention-like mechanism of the ADS architecture which represents the full SCT framework. This mechanism focuses the trackers attention on detections that are on the true location of the object by increasing their sampling distribution since they have a higher heatmap cost. As a result more tracklets are built on the true location of the heatmap which further strengthens the heatmap distribution leading to more detections being sampled and promoted. However this is not the case for the HDS heatmap as it has no influence on the tracker. This results in the tracker creating more false positive tracklets which lead to the strong distribution on the heatmap that also covers non-object areas.

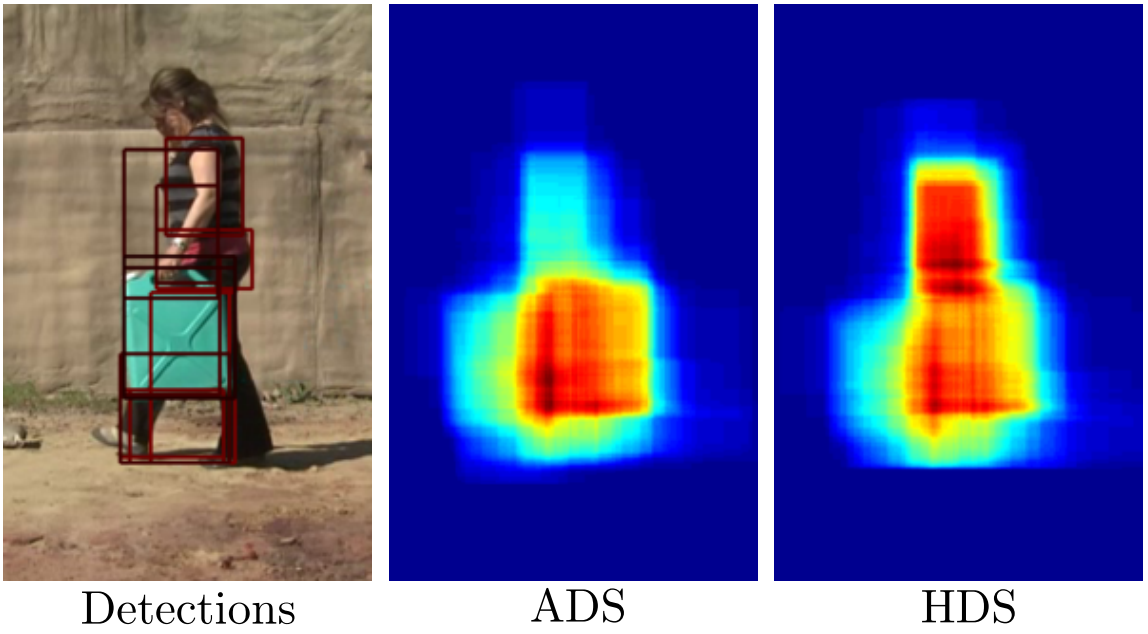


Figure 6.9: Comparison of heatmaps obtained from the ADS and the HDS architectures. The detections image illustrates a reference for the object-person relationships in the heatmaps. Object detections on the image frame are illustrated as red rectangles, where brighter and darker rectangles represent detections with higher and lower costs respectively. We can observe that the heatmap from the ADS approach suppresses the false positives from becoming a strong region in the heatmap due to its attention-like mechanism and only captures the true location of the object, while the strong heatmap distribution of HDS also covers non-object areas.

6.5 Evaluation of Joint Tracking and Event Analysis

In this section we present the evaluation of our Joint Tracking and Event Analysis (JTEA) framework which was described in Chapter 5. To accomplish this, we perform a quantitative and qualitative analysis of the JTEA framework, presented in the following sections.

6.5.1 Quantitative Analysis

In this section we perform a quantitative analysis of the tracking and event analysis aspects of our JTEA framework. The following describes the experimental settings used as part of this quantitative analysis.

6.5.1.1 Experimental Settings

The input of our JTEA framework consists of a set of tracklets, which can be obtained from any tracker, and learnt event models which will be used in an HMM. The following are

the experimental settings describing the process of satisfying these requirements for JTEA and other requirements to obtain the tracking and event analysis results after applying our JTEA framework on the MINDSEYE2015 Dataset.

Tracking

To obtain a set of tracklets, so that we can apply our JTEA framework, we use three trackers: (i) our Spatial Consistency Tracker (SCT) described in Chapter 4, (ii) an unmodified version of Pirsiavash et al. globally-optimal greedy tracker [63], henceforth (DPG), and (iii) an unmodified version of Andriyenko et al. discrete-continuous optimization tracker [4], henceforth (DCO).

For each of the aforementioned trackers, we obtain a set of tracklets as a results of applying them to detections that were obtained after running our carried object detector on the MINDSEYE2015 dataset. We then obtain results for two sets of experiments based on our JTEA framework, represented as *HMM* and *BASE*. The first experiment, *HMM*, applies the full JTEA framework, as described in Chapter 5, on each of the obtained three tracklet sets to obtain object tracks. The tracks obtained through this experiment, for each of the SCT, DPG and DCO trackers, are represented by the *HMM* experiment label in the results section.

The second experiment (*BASE*), applies only the tracking aspect of our JTEA approach in equation 5.2 as its objective function. This experiment effectively ignores the notion of events from JTEA by removing the Gaussian observation term and the event term $P(S|\mathcal{R},\mathcal{T})$ from the objective function. The tracks obtained through this experiment are represented by the *BASE* experiment label. The tracking results of these two experiments provide a means of comparison, to draw conclusions on the effects of jointly using events as part of tracking (*HMM*) against tracking without the influence of events (*BASE*). The tracking results from both experiments are presented in terms of an average $F1$ score across all videos calculated on a frame by frame basis.

We additionally run two sets of evaluations on the aforementioned sets of experiments, using carried object detections obtained from ground truth person tracks (GT) and automatic person tracks (Auto). This is to illustrate that JTEA and also SCT are not heavily dependent on highly accurate person tracks. Our automatic person tracks obtain an average $F1$ score of 0.79, 0.73 and 0.58 with 50%, 60% and 70% overlap thresholds respectively.

Event Analysis

In addition to the experiments based on the tracking performance of our JTEA framework, we also present a quantitative evaluation on the event recognition aspect of JTEA. By

applying our JTEA framework to obtain object tracks, event recognition is simultaneously performed within the HMM using the $P(S|\mathcal{R},\mathcal{T})$ term of Equation 5.2 in Section 5.2. In the MINDSEYE2015 dataset, there are seven types of events in which an object may participate in, relative to two types of reference entities, namely person and scene. These events are *Carry*, *Static* (object is stationary), *Pickup*, *Putdown*, *Drop*, *Raise* and *Roll* (object is moving on the ground).

To train an HMM model and learn the aforementioned events, we take as input ground truth object and person tracks along with a scene bounding box which covers the entire image frame, representing the scene reference track. As described in Section 5.3, we construct an observation matrix for each event based on position and velocity, in the x and y dimensions relative to the two reference entities, person and scene. For each event, we obtain a mean and covariance matrix which we use as event models in our HMM along with prior probabilities of events and a transition matrix constructed from event ground truth. The above HMM training is performed within five folds (one for each recording in the MINDSEYE2015 dataset).

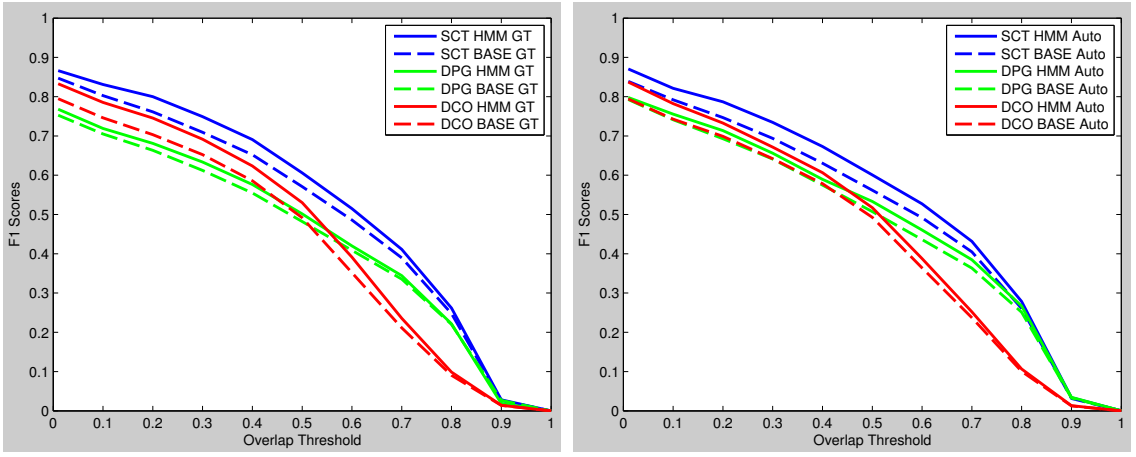
To predict a sequence of events from a test object track, we use the HMM from the fold that the video was not trained on. Here the HMM uses a Viterbi algorithm to predict the most likely sequence of events, given the test object track, person and scene entity tracks (reference objects) and the aforementioned trained model for the HMM. The output of the HMM (in addition to the HMM measure) is a sequence of event labels corresponding to the object track for each frame.

In order to evaluate the sequence of event labels, we present quantitative results of event classification obtained by using the following three experimental settings:

- **HMM Train Test:** This experiment is to evaluate the trained model to be used by the HMM. We perform five-fold cross validation to train and test our HMM on ground truth object and reference object tracks and evaluate the predicted sequence of events against ground truth events based on a frame by frame basis.

- **SCT HMM GT:** This experiment evaluates the events obtained from the HMM using the final tracks generated by the *SCT HMM GT* tracking experiment, against ground truth events. It must be noted that the creation of these tracks were influenced by the events that they produce during the same optimisation.

- **SCT BASE GT:** In this experiment we evaluate the performance of event recognition as a result of using the tracks obtained by the *SCT BASE GT* experiment. The creation of these tracks where not influenced by events and in this experiment we perform event recognition as a post-processing stage.



(a) Ground truth person tracks (b) Automatic person tracks

Figure 6.10: Performance comparison of our JTEA framework. We present two sets of experiments, using ground truth (*GT*) and automatic (*Auto*) person tracks. We apply our three main frameworks to obtain detections, tracklets and tracks using only the person track type in each experiment. For each of the trackers producing tracklets, namely SCT, DPG and DCO, the significant increase in performance of our JTEA framework as a result of incorporating events (HMM) over the baseline (BASE) can be clearly seen.

6.5.1.2 Results & Conclusions

The following presents the results of tracking and event analysis after applying our JTEA framework with the above experimental settings.

Tracking Results

As illustrated in Figure 6.10, no matter which of the three tracklet building trackers were employed, we can conclude that the performance of tracking is significantly improved when influenced by events (*HMM* label), as a result of our joint tracking and event analysis framework, when compared to the *BASE* experiment that does not take events into account and does not take advantage of the object-entity interaction. It must be noted that even a 5% improvement corresponds to approximately 3000 more true positives due to the large number of frames in the dataset.

From the results we can also observe that the tracks obtained by applying the JTEA framework on tracklets provided by our SCT tracker outperforms the tracks obtained by using tracklets from other trackers. We can therefore conclude that the SCT tracker produces significantly improved tracklets for the JTEA framework, compared to the tracklets produced by DPG and DCO. This can be considered as further experiments for Section 6.4, proving that the use of heatmaps as part of our spatial consistency tracker can be of great benefit when creating tracklets.

By comparing the results of Figure 6.10 (a) and (b), we can observe that the object tracking results obtained from ground truth (GT) and automated person tracks (Auto) are very similar. We can therefore conclude that our JTEA and SCT approach do not heavily rely on high quality person tracks and are robust against noisy person tracks. It is also worth highlighting how the performance of the system does not drop rapidly for higher values of overlap threshold, showing the potential of our carried object detector in localising objects accurately.

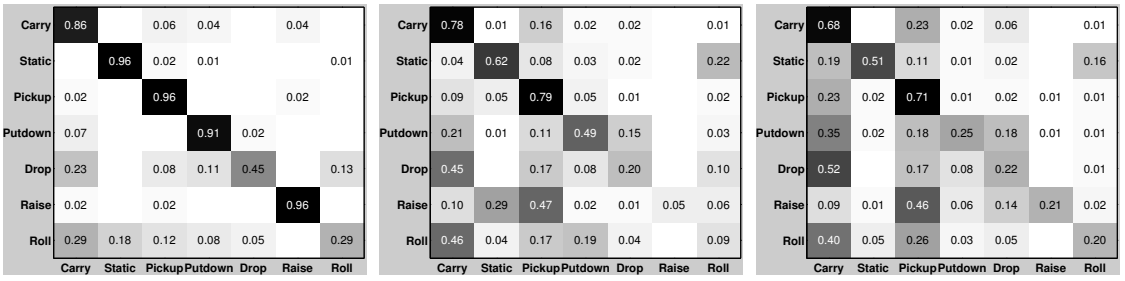
For a more in depth comparison with related works, in Table 6.2 we provide a summary of performance indexes computed at 20% overlap, a value typically employed in the literature for carried object tracking [24, 28]. Since there is no major difference between the GT and Auto results, we only provide detailed information for the GT (Figure 6.10 (a)) evaluation.

	$F1$	Precision	Recall	Accuracy	Run Time
SCT HMM	**0.80**	**0.81**	**0.79**	**0.67**	$<$ 25 min
SCT BASE	0.76	0.77	0.75	0.61	$<$ 25 min
DCO HMM	0.75	0.76	0.73	0.59	$<$ 5 min
DCO BASE	0.70	0.72	0.69	0.54	$<$ 5 min
DPG HMM	0.68	0.69	0.67	0.52	$<$ 1 min
DPG BASE	0.66	0.67	0.66	0.50	$<$ 1 min

Table 6.2: Performance indexes of carried object tracks from GT person tracks, evaluated against $> 20\%$ overlap with ground truth.

We can again verify that for all the different indexes, event recognition always improves the tracking performance and that the SCT tracker outperforms other trackers. Since the notion of object-entity interaction is incorporated in both the SCT tracker and the JTEA framework (HMM), to accurately measure the benefits of incorporating the aforementioned interaction within tracking, we must compare the *SCT HMM* row of the table to the *DCO BASE* and the *DPG BASE* rows, as they do not use any notion of interaction, which corresponds to a 10% and 14% improvement in $F1$ score respectively. This significant improvement is purely as a result of using interaction within tracking in our *SCT HMM* approach.

The run times in Table 6.2 are for the trackers only, and that JTEA ran an additional 5 minutes for each tracker. All given times are calculated using a single core on an Intel Xeon E5-2665 Processor @2.40GHz. Although SCT is slower than DCO and DPG, we believe it can be run in a more comparable time if optimised.



(a) HMM Train Test (b) SCT HMM GT (c) SCT BASE GT

Figure 6.11: Confusion matrices for event classification.

Event Analysis Results

Figure 6.11 presents the confusion matrices produced as a result of evaluating the event recognition procedure for each of the three experiments outlined in section 6.5.1.1. The confusion matrix in Figure 6.11 (a) illustrates that the HMM approach to modelling events for carried objects is suitable for the problem. This also highlights the suitability of the object-entity relations used as observations in modelling the HMM.

Confusion matrices in (b) and (c) show that HMM based event classification notably improves the baseline, further clarified by the number of correct event classifications reported in Table 6.3. Based on this table, the ground truth based classification in (a) allows for consistently better results in all classes, but the HMM event recognition substantially improves over the baseline's performance leading to 64.2% vs. 53.9% correct classifications respectively.

It must be noted that many of the false positive event classifications are due to the nature of the frame by frame evaluation where event intervals may be slightly misaligned with the ground truth, leading to a larger number of false positives which may in fact be considered as true positives. This is particularly the case for very short temporally occurring events such as *drop* and *raise*.

	Carry	Static	Pickup	Putdown	Drop	Raise	Roll	Total	Total %
Sum GT Frames	10787	36578	1402	1521	83	530	259	51160	100
HMM Train Test	9559	35337	1340	1379	37	508	74	48234	94.3
SCT HMM	8382	22537	1108	742	17	28	24	32838	64.2
SCT BASE	7398	18630	994	375	18	113	53	27581	53.9

Table 6.3: Total number of true positive event detections based on a frame by frame evaluation.

6.5.2 Qualitative Analysis

In this section we present a qualitative analysis of the tracking results provided by the *SCT HMM* and *SCT BASE* experiments on the MINDSEYE2015 dataset, by comparing the quality of their produced tracks and additionally highlighting the benefits of incorporating events into the tracking process. The experimental settings for this analysis is described below.

6.5.2.1 Experimental Settings

Figure 6.12 illustrates the tracking results of *SCT HMM* and *SCT BASE* for two different cases, namely (a) and (b). The rows of each case represent sample image frames that are temporally ordered. Between each of the ordered frames, gaps of multiple frames may exist. In each frame the person bounding box is represented by a black bounding box, ground truth object track as green, the *SCT HMM* object track as red and the *SCT BASE* as blue. For each of the object tracks the corresponding event recognition in that frame is displayed in the gray box at the top left corner of each image frame. The green text represents the ground truth event of the green ground truth track, the red text represents the *SCT HMM* event for the red object track and the blue text represents the *SCT BASE* event for the blue object track.

While the green ground truth events where manually annotated, the red events for *SCT HMM* where obtained within the JTEA optimisation while tracking, whereas the blue events for *SCT BASE* where obtained using the HMM in a post-processing stage after the *BASE* tracking process had finished.

6.5.2.2 Results & Conclusions

Figure 6.12 illustrates tracking results by focusing on the detections within the tracks. We can observe that in both cases (a) and (b), the red object tracks of *SCT HMM* more accurately cover the green ground truth object track, compared to the blue object tracks of *SCT HMM*. Moreover, the events of the red *SCT HMM* approach is also more accurate than the blue *SCT BASE* events. This improvement in tracks and event recognition is a direct result of incorporating events within the tracking process. The influence of events in improving the tracks for each case is described below.

In the first case, (a), the person drops the object after carrying it. Since the *SCT HMM* detects a *drop* event, the red object track of *SCT HMM* has a trajectory that conforms to a *drop* event, that is it has a downward direction with a higher than normal velocity. Due to the knowledge of this event, the red object track accurately follows the dropped object.

However, for the blue *SCT BASE* object track, since it does not use any notion of events and does not have any knowledge of what is happening, it continues to track the object as it is still being carried, which is confirmed by the post-processed blue event recognition.

The second case, (b), illustrates an example where an object is carried, but is highly occluded. In this example the red object track of *SCT HMM* continues to follow the ground truth object track, even in frames where it is highly occluded. We can observe that the red object track has this behaviour as the event *carry* is correctly predicted which enforces the trajectory of the red object track to have a consistent spatial behaviour with respect to the entity possessing it. In this example, the tracking procedure may prefer using interpolated detections as part of the trajectory of the object track, since they more accurately cover the true location of the object compared to partial or false positive object detections. The choice of whether to use interpolated or other detections which may be false positives is determined based on the HMM measure in the JTEA objective function, which is dependant on the event sequence that the trajectory produces in the HMM. The importance of this is highlighted when we compare against the blue *SCT BASE* object track which is not constraint by and does not follow a particular event model. It therefore uses false positive detections to create its final trajectory rather than interpolating, which we can observe in the post-processed event recognition as an incorrect *pickup* event.

Figure 6.13 illustrates example tracking results by presenting the full trajectories of *SCT HMM* and *SCT BASE*. We can observe that our *SCT HMM* trajectories more accurately follow the green ground truth trajectories compared to the blue *SCT BASE* trajectories. They are also smoother due to the events they produced and were influenced by. In Figure 6.13 (a) however, the detector produces detections that cover both the carried object and the feet and as a result our red trajectory, which follows the centre of the detection bounding boxes, is lower than that of the ground truth. In Figure 6.13 (b) we can observe how how our red trajectories continues to follow the object even when it is fully occluded.

Figure 6.14 presents additional examples of trajectories in a different and more challenging viewpoint. In Figure 6.14 (a) we can observe when a second object is present in the scene (a flag), by taking advantage of the knowledge of events, our red *SCT HMM* trajectory does not alternate between the two objects and persists to remain on one. In Figure 6.14 (b) we can observe that when the object is picked up, due to high occlusion both *SCT HMM* and *SCT BASE* initially lose the object. However our approach finds the object again and using the knowledge of the event *Carry* follows the true location of the object.

Figure 6.15 presents another set of examples based on the same object but in two different viewpoints. In Figure 6.15 (a), due to the way the object is carried, the object has

a very narrow profile and is challenging to detect and track. However, in Figure 6.15 (b) which captures the object from a different viewpoint, the object is more visible and easier to track.

In the next section we present an overall conclusion on the evaluation and experiments presented in this chapter.

6.6 Overall Conclusions

In this chapter we performed quantitative and qualitative evaluations of the three frameworks outline in this thesis, namely our Geometric Carried Object Detector, our Spatial Consistency Tracker and our Joint Tracking and Event Analysis frameworks.

We initially evaluated our carried object detector and showed the significantly improved performance it obtained over other state-of-the-art approaches with respect to its object detections.

We then evaluated and showed the benefits of incorporating spatial consistency within the tracklet building process of our spatial consistency tracker. Our tracker's ability to suppress false positives and maximise the number of true positive in the scene based on spatial consistency gives it an advantage over other trackers.

Finally, and most importantly, we evaluated our joint tracking and event analysis framework. In this evaluation we illustrated the benefits of this framework which accomplishes the main goal of this thesis, that is, to use interaction as a type of context and incorporate it within tracking. We can conclude from the results of this evaluation that we can greatly benefit from using high level notions of interactions within object tracking.

In the next chapter we provide an overall conclusion on the work presented in this thesis and provide insights into future directions.

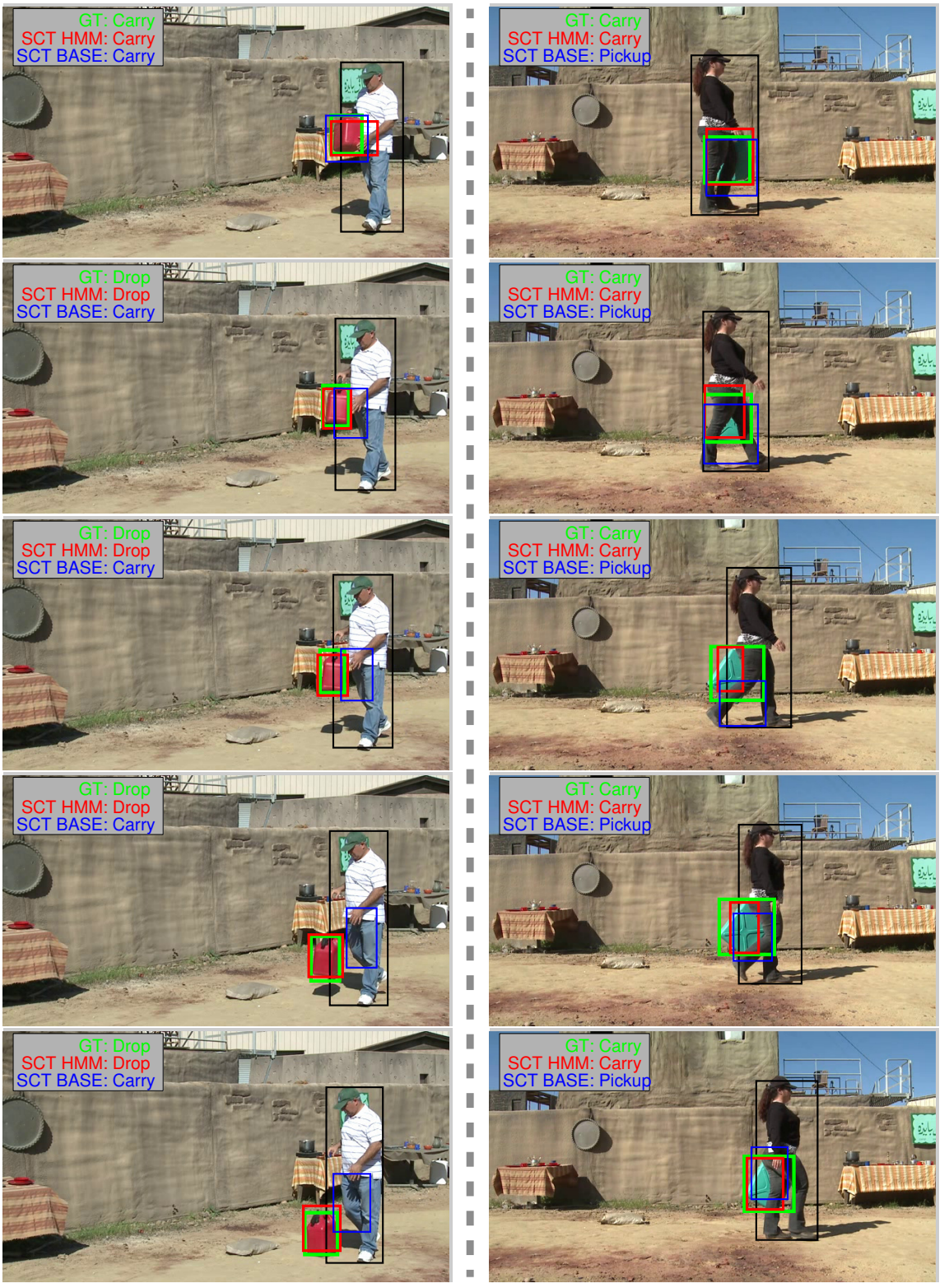


Figure 6.12: Qualitative analysis of the tracking and event recognition results of *SCT HMM* and *SCT BASE* at a detection level. Each column represents a different example of the aforementioned results. The rows of each column illustrate a sequence of frames where the red *SCT HMM* object tracks and events outperform the blue *SCT BASE* object tracks and events, when compared to the green ground truth. This improvement in tracking and event recognition is due to incorporating events in the tracking process, which can be concluded by analysing the tracks and their corresponding events.

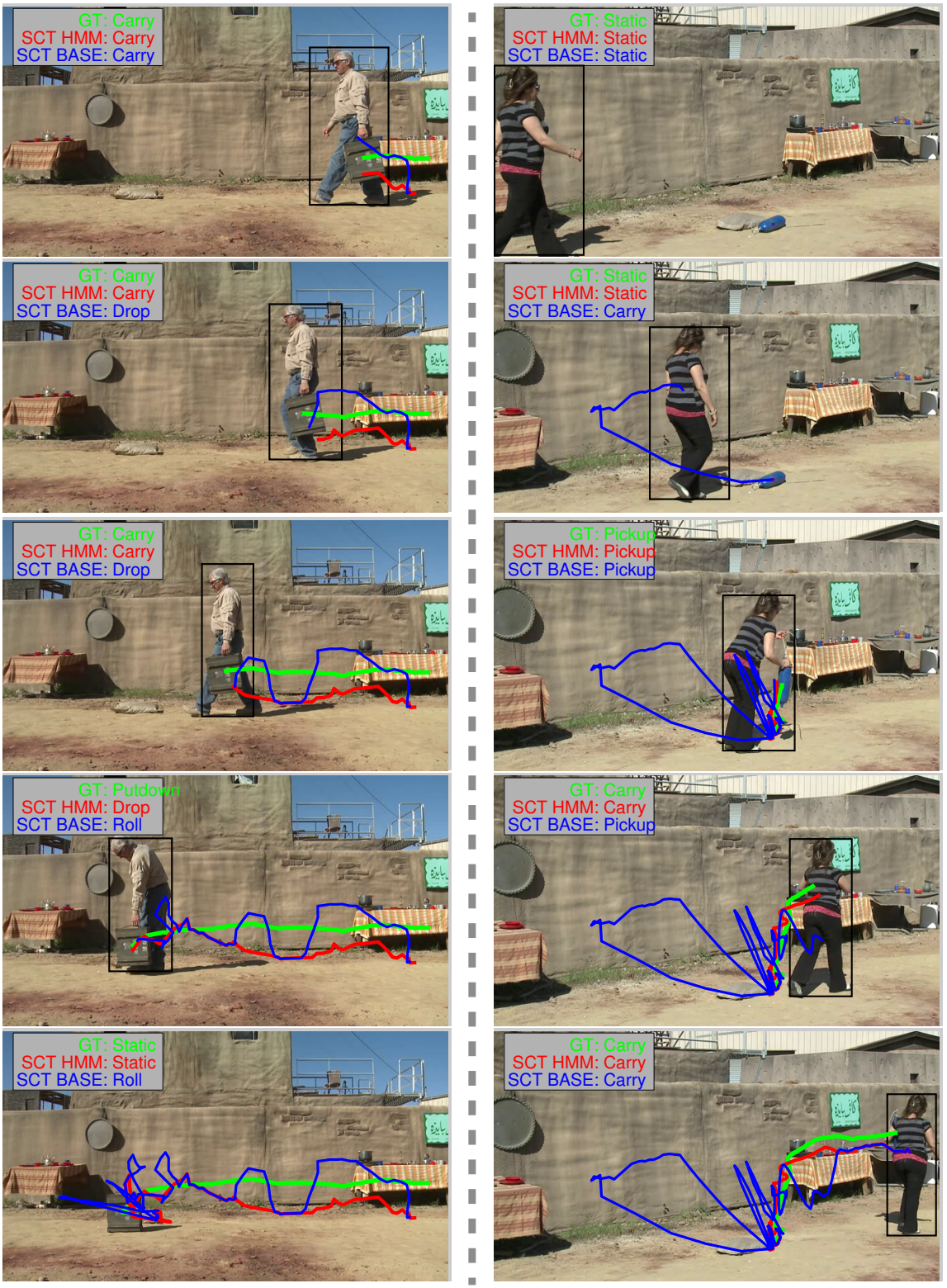


Figure 6.13: Qualitative analysis of the trajectories and event recognition results of *SCT HMM* and *SCT BASE*. We can observe that our *SCT HMM* trajectories more accurately follow the green ground truth trajectories compared to the blue *SCT BASE* trajectories.

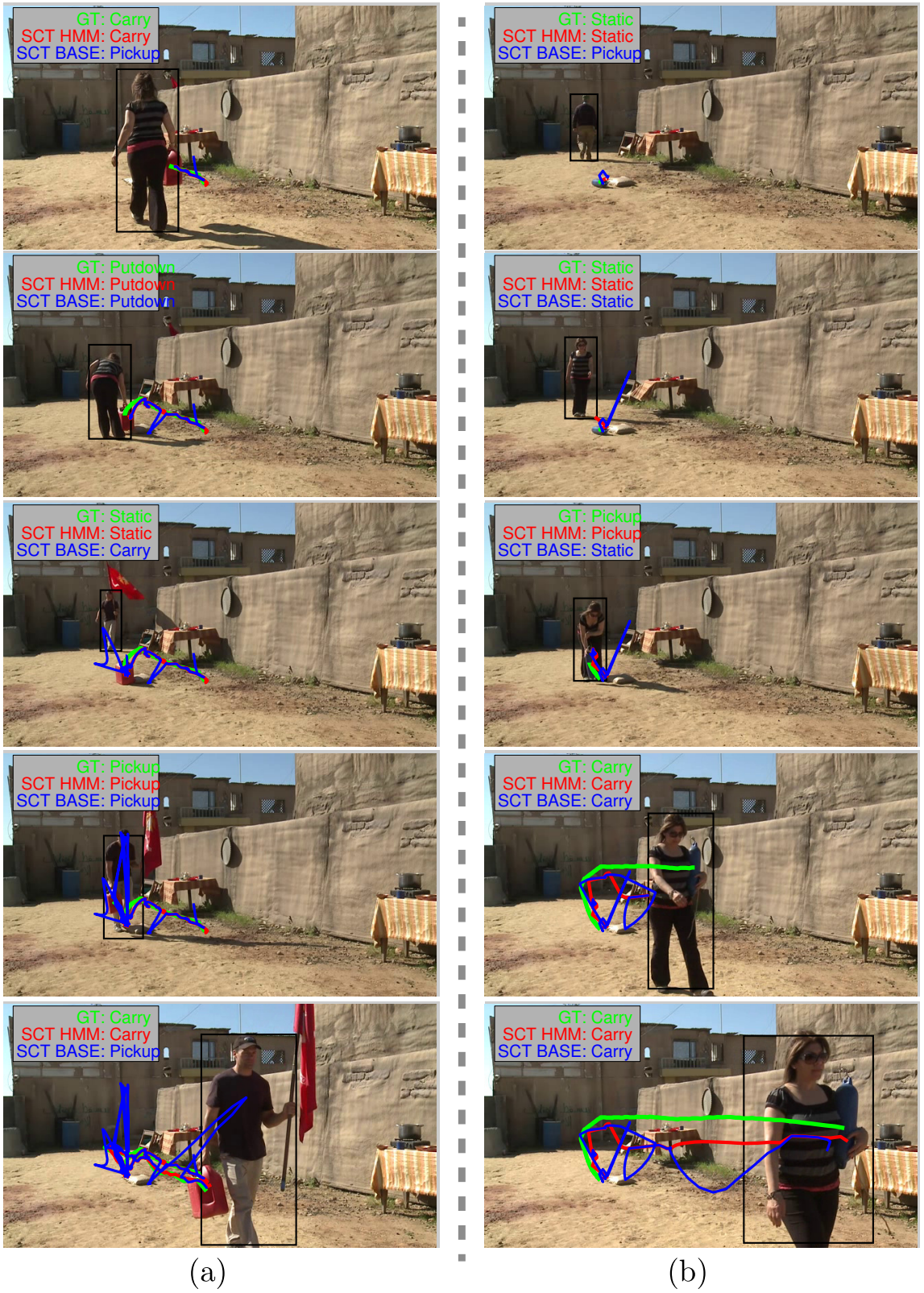


Figure 6.14: Qualitative analysis of additional trajectories and event recognition results of *SCT HMM* and *SCT BASE*. We can observe that our *SCT HMM* trajectories more accurately follow the green ground truth trajectories compared to the blue *SCT BASE* trajectories.

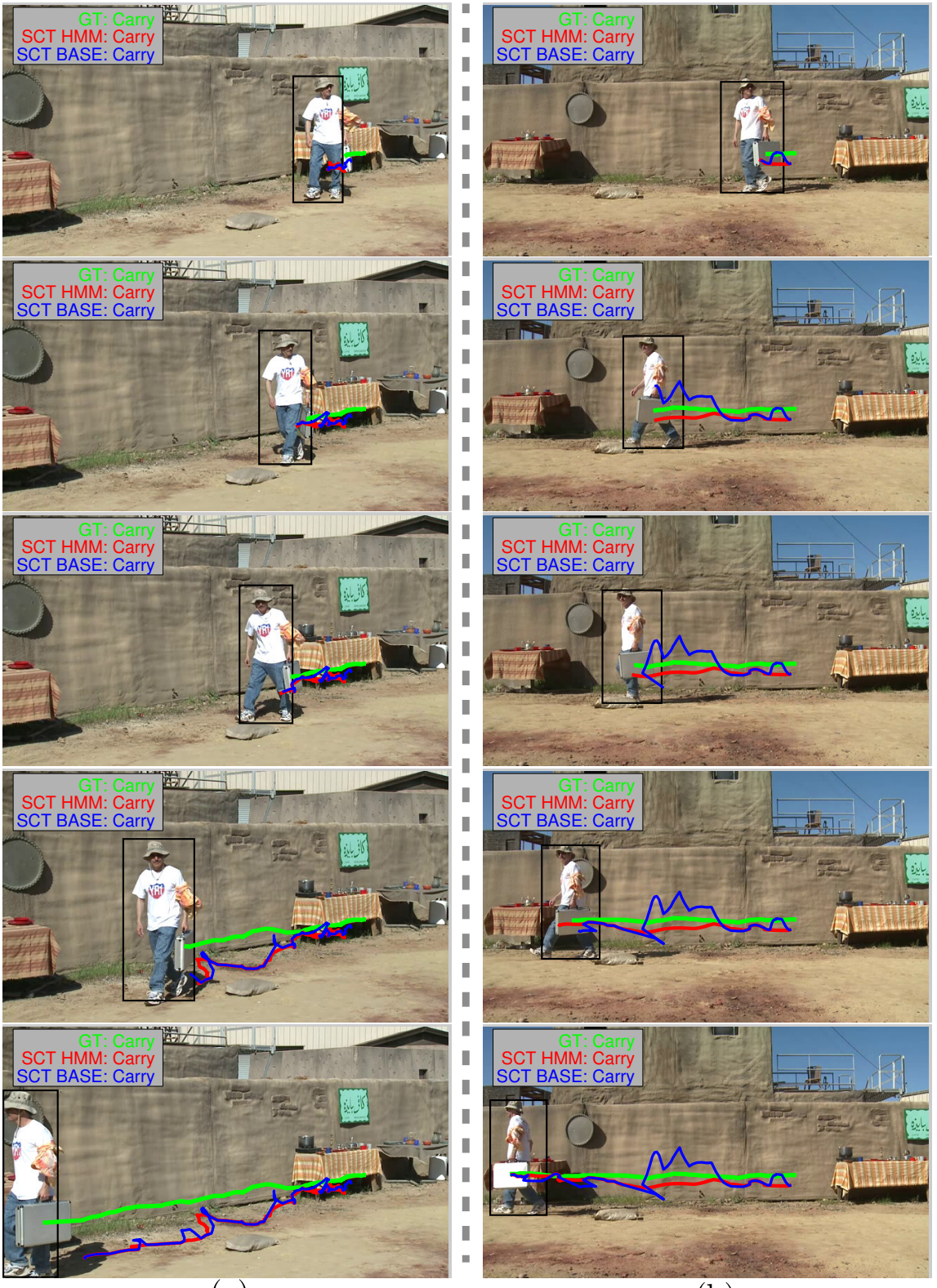


Figure 6.15: A good and bad example of the same object in different viewpoints. In (a), due to the way the object is carried, the object has a very narrow profile and is challenging to detect and track. However, in (b) which captures the object from a different viewpoint, the object is more visible and easier to track.

Chapter 7

Conclusion and Future Work

This thesis investigated the problem of tracking in the context of interaction. We presented our approach to this problem in three different parts of the thesis, (i) our geometric carried object detector, (ii) detection and tracking through spatial consistency and (iii) joint tracking and event analysis; each of these novel frameworks are briefly described below.

In Chapter 3 we presented the area in which we would like to investigate the effects of incorporating interaction within the tracking process, that is, the area of carried objects. We provided a full literature review of this area in Section 2.1 and highlighted the many challenges within this domain; challenges that have been the main reason there has been relatively little work in this area. We therefore presented our geometric carried object detector in Chapter 3 that produces detections and provides a basis where we can apply our trackers that incorporate the knowledge of interactions.

In Chapter 4 we used the obtained detections and locally connect them to form tracklets using our spatial consistency tracker framework. Here, we exploited the spatial consistency between the object and the entity interacting with it to obtain better tracklets that partly cover the trajectory of the object. The notion of spatial consistency captures certain interactions where the trajectories of the object and entity follow the same behaviour and are consistent with respect to one another. We built tracklets in this manner before performing object tracking so that we can take advantage of the notion of interactions at a detection level, where a detection was modelled with respect to a single entity at each frame.

Chapter 5 presents our joint tracking and event analysis framework (JTEA). In this

chapter we tackled and solved the main goal of this thesis, tracking in the context of interaction. Here we performed tracking by taking into account modelled interactions based on spatial consistency and inconsistency where in each frame, the target object is tracked based on its relationship with respect to all entities in the scene. After obtaining interactions in this manner, they influenced and improved object tracking which in turn produced improved knowledge of interactions.

7.1 Contributions

7.1.1 Carried object detection

We have introduced a novel approach to carried object detection by providing a vision system that detects a large variety of carried objects. Our approach characterises carried objects in terms of generic shape properties such as convexity, whilst taking advantage of the fact that they are often, but not always, protruding from a person silhouette.

Based on the evaluations in sections 6.4.2 and 6.4.1, our object detector provides detections that better localise the carried objects when compared to other state-of-the-art approaches. Our detector does not require camera settings and works for a variety of camera angles and viewpoints.

7.1.2 Tracking through spatial consistency

We introduced a tracking framework that exploits the continuous and spatially consistent relationship that object trajectories have relative to the entity interacting with them. In addition, an iterative event driven optimisation process which incorporates a heatmap and an attention-like mechanism is used to obtain an optimal set of object tracklets.

Experimental results in section 6.4.1 and 6.4.2 show that our approach significantly outperforms other state-of-the-art techniques, especially highlighting the benefits of iteratively updating the heatmap, promoting low strength true positives and its attention-like mechanism, all of which influence the tracking process.

7.1.3 Joint tracking and event analysis

Despite the increasing efforts put by the computer vision community into tracking objects and people from videos, few approaches have investigated the benefit of performing tracking jointly with event analysis. We presented a novel approach to the problem of tracking objects which various entities may interact with in different ways. We model

interaction as events using a Hidden Markov Model and incorporate it into a joint tracking and event analysis optimisation which selects from the optimal subset of tracklets and forms the final carried object track.

Based on the evaluations performed and the results provided in section 6.5, we have shown that the inclusion of event analysis in the optimisation process significantly improves the tracking performance. This improvement was consistent across our spatial consistency tracker and two other employed state-of-the-art trackers. We have also shown that event classifications which were simultaneously obtained with object tracks substantially improve when using our JTEA framework. This improvement in events is due to the improvement in the tracks where the mutual influence and improvement between tracking and event analysis is at the core of our novel joint tracking and event analysis framework. Moreover, the tracking results of our spatial consistency tracker outperform the other two trackers, which is due to its robustness to false positives as a result of using spatial consistency between the object and the possessing entity.

7.2 Future Work

While various improvements and extensions may be applied to each of the aforementioned previous frameworks, for example extending our approach to include multi-person, multi-object events such as giving, following, exchanging or replacing objects, in this section we present two main directions as part of future work.

7.2.1 HMM based tracking and event analysis

In our joint tracking and event analysis framework we performed tracking within an optimisation procedure where event analysis, performed using a Hidden Markov Model (HMM), jointly influenced the tracking solution. However, it would be interesting to investigate the possibility of further combining both tracking and event analysis within a single HMM-like architecture. To accommodate for this, we would change the architecture illustrated in Figure 5.1, a standard HMM architecture which takes tracks as observations and outputs event sequences, and propose the architecture illustrated in Figure 7.1. This architecture represents a switched Kalman filter which takes detections as input (bottom layer) and performs tracking (middle layer) and event analysis (top layer) by automatically switching between states, or in this case modelled interactions.

This notion of model-switching was originally introduced by Isard and Blake [42] where various motion models were used for tracking, e.g. a bouncing ball or gesture

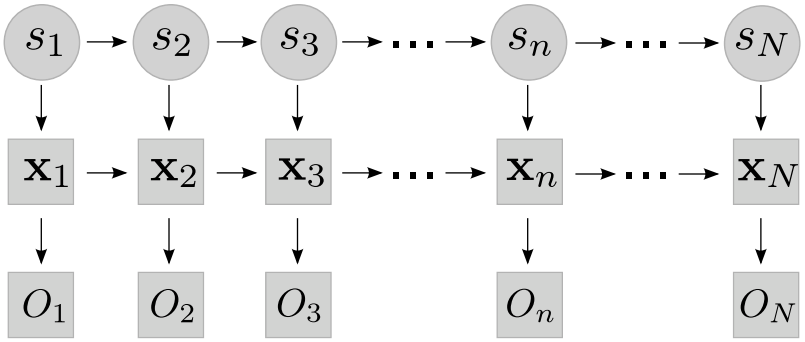


Figure 7.1: A proposed switched Kalman filter architecture which would perform both tracking and event analysis under one architecture. The bottom layer is for detection observations, the middle layer represents the layer and the top layer represents the events where N is the total number of frames.

recognition, where each model constrains the behaviour of a trajectory differently, leading to a significant improvement of the tracking process. We use a similar idea in our proposed switched Kalman filter where the detections form trajectories using a Kalman filter in the middle layer; however similar to our joint tracking and event analysis framework, event models are switched at the top layer, influencing and improving the trajectory constructed by the Kalman filter.

There is however an additional challenge introduced at the detection layer where a frame may not have any detections, or detections in a particular frame may not contain any true positives. While our joint tracking and event analysis framework handled this situation by interpolating through such frames, a similar approach such as Kalman smoothing is needed to handle such cases.

7.2.2 Mutual influence between frameworks

The heatmap constructed during our Spatial Consistency Tracker (SCT) models the spatially consistent relationship between an object and the entity interacting with it, for example during a *carry* event. However in our current SCT framework, there is no notion of high level events (such as the ones detected in the JTEA approach) which define intervals that describe whether the object is spatially consistent with respect to the person or if it is inconsistent. Therefore, the heatmap is constructed for the entire duration an entity is present and continues to be updated even if the object is no longer spatially consistent with respect to the person, e.g. it is dropped or put down.

This is not a problem for the heatmap since tracklets that are during spatially inconsistent intervals are relatively sporadic and do not emerge as strong distributions within

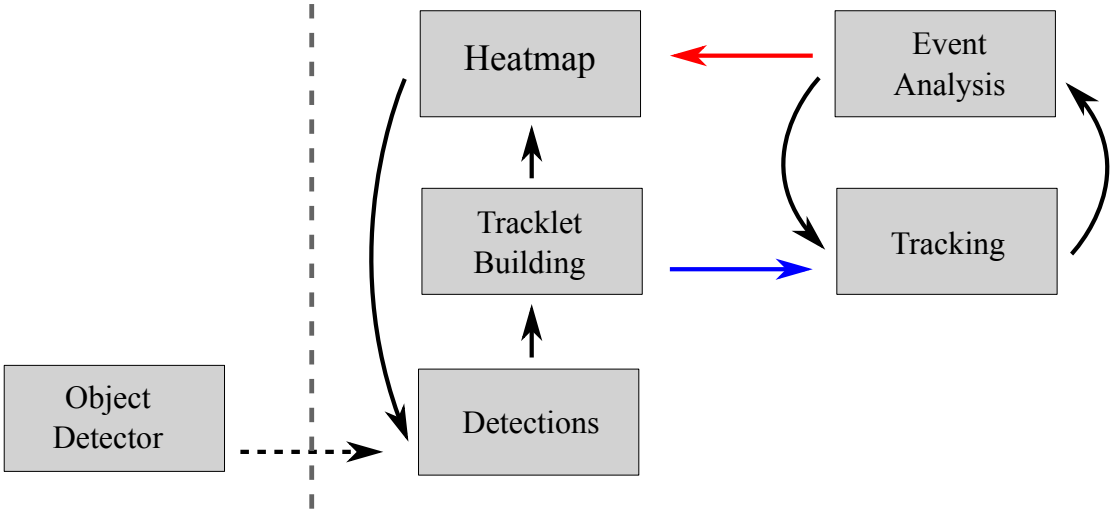


Figure 7.2: A proposed approach to combine our Spatial Consistency Tracker (SCT) and our Joint Tracking and Event Analysis (JTEA) frameworks. The new influence indicated by a red arrow allows the knowledge of obtained events from JTEA to improve the quality of the heatmaps in SCT, subsequently improving the tracklets in SCT which, as influenced by the blue arrow, will improve the quality of the tracking process in JTEA.

the heatmap. However, it would be very beneficial to construct the heatmap only during intervals where the entity is present and only when the object has a spatially consistent behaviour with respect to the entity, rather than during the entire duration where the entity is present. This would provide a much more accurate heatmap. Since we obtain event sequences in our JTEA framework, which can define intervals where the object is spatially consistent or not, it would be interesting to combine our SCT and JTEA frameworks to obtain more accurate heatmaps, which in turn may provide more accurate tracklets. We propose an architecture combining the two frameworks, where we modify Figure 1.2 which consists of the different frameworks and provide Figure 7.2 illustrating our proposed combined architecture.

In Figure 7.2, the red arrow introduces a new influence where the knowledge of events obtained after applying JTEA can be used to improve the quality of the heatmaps in SCT. As a result of this improvement, tracklets in SCT may also improve which directly influence the tracking process in JTEA, indicated by a blue arrow, which may also lead to an improvement in the quality of tracks produced by the tracking process.

The main challenge in combining the two frameworks is the way they are to be applied. We propose two main ways, firstly one can apply both frameworks simultaneously and

iteratively. In this approach tracklets and tracks are iteratively constructed including their corresponding heatmaps and events. The alternative second approach is to complete the SCT aspect of the combined framework, use the tracklets to complete the JTEA aspect of the combined framework, and then proceed with running the SCT aspect again but this time with the knowledge of obtained events from JTEA. This will result in more accurate heatmaps and improved tracklets. After SCT is completed again we move on to completing JTEA using the new and improved tracklets. This process may continue until a certain number of cycles has been completed. Deciding which of these approaches is better suited for the task of combining SCT and JTEA would be one of the main goals of this future work.

7.3 Closing Remarks

This thesis has introduced a framework where tracking and event analysis were performed jointly, where they mutually influenced and improved each other. We also presented a framework incorporating notions of events at a detection level in our spatial consistency tracker. This tracker builds tracklets using detections from our geometric carried object detector. Based on quantitative and qualitative results, we experimentally validated the hypothesis that events can have a significant role in improving object tracking. While this work is a small but important step towards incorporating event analysis within computer vision problems, we hope to see a larger trend where others take advantage of using high-level knowledge to improve lower-level problems.

Bibliography

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of 20th Intl. Conf. on VLDB*, pages 487–499, 1994.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[3] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *European Conference on Computer Vision (ECCV)*, pages 466–479, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[4] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, pages 1926–1933. IEEE, 2012.

[5] A. Barbu, N. Siddharth, A. Michaux, and J. M. Siskind. Simultaneous object detection, tracking, and event recognition. *Advances in Cognitive Systems*, 2:203–20, 2012.

[6] C. BenAbdelkader and L. S. Davis. Detection of people carrying objects: A motion-based recognition approach. In *IEEE International Conference on Automatic Face and Gesture Recognition FGR*, pages 378–383. IEEE Computer Society, 2002.

[7] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8, Dec 2009.

[8] N. Bergman and A. Doucet. Markov chain monte carlo data association for target tracking. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 2, pages II705–II708, 2000.

[9] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi. Examiner: optimized level-wise frequent pattern mining with monotone constraints. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 11–18, Nov 2003.

[10] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, Sept 2004.

[11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, November 2001.

[12] A. Branca, M. Leo, G. Attolico, and A. Distante. Detection of objects carried by people. *Proc. Intl Conf. Image Processing*, 3:317–320, 2002.

[13] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1820–1833, Sept 2011.

[14] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(1):90–99, Jan 1986.

[15] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.

[16] R. Chayanurak, N. Cooharojananone, S. Satoh, and R. Lipikorn. Carried object detection using star skeleton with adaptive centroid and time series graph. In *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, pages 736–739, Oct 2010.

[17] Wongun Choi and Silvio Savarese. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision*, chapter A Unified Framework for Multi-target Tracking and Collective Activity Recognition, pages 215–230. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[18] C. Chuang, J. Hsieh, L. Tsai, S. Chen, and K. Fan. Carried object detection using ratio histogram and its application to suspicious event analysis. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(6):911–916, June 2009.

[19] C. Chuang, J. Hsieh, L. Tsai, and K. Fan. Human action recognition using star templates and delaunay triangulation. In *Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, pages 179–182, 2008.

[20] J.B. Collins and J.K. Uhlmann. Efficient gating in data association with multivariate gaussian distributed states. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(3):909–916, Jul 1992.

[21] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149 vol.2, 2000.

[22] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):781–796, Aug 2000.

[23] D. Damen and D. Hogg. Detecting carried objects in short video sequences. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pages 154–167, Berlin, Heidelberg, 2008. Springer-Verlag.

[24] D. Damen and D. Hogg. Detecting carried objects from sequences of walking pedestrians. *PAMI*, 34(6):1056–1067, 2012.

[25] DARPA. Mind's eye challenge http://www.visint.org/. `http://www.visint.org/`, 2011.

[26] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1):1–27, 2011.

[27] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, June 2009.

[28] R. Dondera, V. Morariu, and L. Davis. Learning to detect carried objects with minimal supervision. In *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, pages 759–766, June 2013.

[29] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *STATISTICS AND COMPUTING*, 10(3):197–208, 2000.

[30] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and Ramanan D. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.

[31] A. Friedman. Framing pictures: the role of knowledge in automatized encoding and memory for gist. *J Exp Psychol Gen*, 108(3):316–55, 1979.

[32] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, pages 1–8, june 2008.

[33] A. Gilbert and R. Bowden. *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II*, chapter Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns of Activity, pages 125–136. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[34] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

[35] S. Godsill, A. Doucet, and M. West. Maximum a posteriori sequence estimation using monte carlo particle filters. *Annals of the Institute of Statistical Mathematics*, 53(1):82–96, 2001.

[36] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson/Prentice Hall, 2008.

[37] I. Haritaoglu, R. Cutler, D. Harwood, and L. S. Davis. Backpack: Detection of people carrying objects using silhouettes. *Computer Vision and Image Understanding*, 81(3):385 – 397, 2001.

[38] G. Heitz and D. Koller. Learning Spatial Context: Using Stuff to Find Things. In *ECCV*, volume 5302, chapter 4, pages 30–43. Springer, 2008.

[39] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *International Journal on Computer Vision*, 80(1):3–15, October 2008.

[40] D. Hoiem, A.A. Efros, and M. Hebert. Geometric context from a single image. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 654–661 Vol. 1, Oct 2005.

[41] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision (ECCV)*, volume 1, pages 343–356, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[42] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *Computer Vision, 1998. Sixth International Conference on*, pages 107–112, Jan 1998.

[43] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In Toms Pajdla and Ji Matas, editors, *ECCV*, volume 3024 of *Lecture Notes in Computer Science*, pages 279–290. Springer Berlin Heidelberg, 2004.

[44] H. Kinoshita. Effects of different loads and carrying systems on selected biomechanical parameters describing walking gait. *Ergonomics*, 28(9):1347–1362, 1985.

[45] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, pages 65–81. Springer-Verlag, 2002.

[46] P. Kovesi. Matlab and octave functions for computer vision and image processing. `http://people.csse.uwa.edu.au/pk/Research/MatlabFns/index.html`, 2006.

[47] D. Kroon and C. H. Slump. Coherence filtering to enhance the mandibular canal in cone-beam ct data. In *Proceedings of the 4th Annual Symposium of the IEEE-EMBS Benelux Chapter*, pages 41–44, 2009.

[48] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, pages 1284–1291, 2005.

[49] C. Lee and A. Elgammal. Carrying object detection using pose preserving dynamic shape models. In *Articulated Motion and Deformable Objects*, volume 4069 of *Lecture Notes in Computer Science*, pages 315–325. Springer Berlin Heidelberg, 2006.

[50] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[51] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1683–1698, Oct 2008.

[52] L. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *IEEE*, pages 1 –8, oct. 2007.

[53] Y. Li and R. Nevatia. Key object driven multi-category object recognition, localization and tracking using spatio-temporal context. In *ECCV*, pages 409–422. Springer, 2008.

[54] X. Liu, L. Lin, S. Yan, H. Jin, and W. Tao. Integrating spatio-temporal context with multiview representation for object recognition in visual surveillance. *Circuits and Systems for Video Technology,*, 21(4):393 –407, april 2011.

[55] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

[56] M. Minoh, T. Yamashita, and K. Ikeda. Automated reforming of an on-line rough sketch based on perceptual organization. In *6th IFSA World Congress*, volume 1, pages 661–664, July 1995.

[57] H. Nanda, C. Benabdelkedar, and L. Davis. Modelling pedestrian shapes for outlier detection: a neural net based approach. In *IEEE Intelligent Vehicles Symposium*, pages 428–433, June 2003.

[58] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 735–742 Vol.1, Dec 2004.

[59] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for multi-target tracking. *Transactions on Automatic Control,*, 54(3):481–497, March 2009.

[60] K. Okuma, A. Taleghani, N. Freitas, J. J. Little, and D. G. Lowe. *Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, chapter A Boosted Particle Filter: Multitarget Detection and Tracking, pages 28–39. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[61] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[62] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *International joint conference on artificial intelligence(IJCAI*, pages 1160–1171, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[63] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201–1208, 2011.

[64] Y. Qi, G. Huang, and Y. Wang. Carrying object detection and tracking based on body main axis. In *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on*, volume 3, pages 1237–1240, Nov 2007.

[65] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[66] N. Rasiwasia and N. Vasconcelos. Holistic context models for visual recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):902–917, May 2012.

[67] J. Roberts. *The Oxford dictionary of the classical world*. Oxford University Press, 2008.

[68] J.A. Roecker. A class of near optimal jpda algorithms. *Aerospace and Electronic Systems, IEEE Transactions on*, 30(2):504–510, Apr 1994.

[69] B. C. Russell, A. Torralba, A. Oliva, and W. T. Freeman. Object recognition by scene alignment. *Advances in Neural Information and Processing Systems*, 2007.

[70] V. Salari and I.K. Sethi. Feature point correspondence in the presence of occlusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(1):87–91, Jan 1990.

[71] T. Senst, R. Heras E., V. Eiselein, M. Ptzold, and T. Sikora. Towards detecting people carrying objects - a periodicity dependency pattern approach. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, pages 524–529, 2010.

[72] T. Senst, R.H. Evangelio, and T. Sikora. Detecting people carrying objects based on an optical flow motion model. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 301–306, Jan 2011.

[73] T. Senst, A. Kuhn, H. Theisel, and T. Sikora. Detecting people carrying objects utilizing lagrangian dynamics. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 398–403, Sept 2012.

[74] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, Jun 1994.

[75] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object. In *ECCV*, pages 1–15, 2006.

[76] J. Mark Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 347–360, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[77] S. Stalder, H. Grabner, and L. Van Gool. Cascaded confidence filtering for improved tracking-by-detection. In *ECCV*, pages 369–382. Springer, 2010.

[78] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1371–1375, Dec 1998.

[79] R. L. Streit and T. E. Luginbuhl. Maximum likelihood method for probabilistic multihypothesis tracking. In *In Proceedings of the International Society for Optical Engineering (SPIE)*, volume 2235, pages 394–405, 1994.

[80] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference on Computer Vision (ECCV)*, volume 6311 of *Lecture Notes in Computer Science*, pages 438–451. Springer Berlin Heidelberg, 2010.

[81] L. Szathmary and A. Napoli. A.: Coron: A framework for levelwise itemset mining algorithms. In *In: Suppl. Proc. of ICFCA 05*, pages 110–113, 2005.

[82] D. Tao, X. Li, S. J. Maybank, and X. Wu. Human carrying status in visual surveillance. In *CVPR*, pages 1670–1677. IEEE Computer Society, 2006.

[83] A. Tavanai. Geometric carried object detector. `https://github.com/AryanaTavanai/Geometric-Carried-Object-Detector`, 2015.

[84] A. Tavanai. Mindseye2015: Joint tracking and event analysis for carried object detection. `http://doi.org/10.5518/9`, 2015.

[85] A. Tavanai, M. Sridhar, Feng Gu, A.G. Cohn, and D.C. Hogg. Context aware detection and tracking. In *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)*, pages 2197–2202, Washington, DC, USA, Aug 2014. IEEE Computer Society.

[86] A. Torralba. Contextual Priming for Object Detection. *International Journal on Computer Vision*, 53(2):169–191, July 2003.

[87] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2 of *ICCV '03*, pages 273–280, Washington, DC, USA, 2003. IEEE Computer Society.

[88] K. Tsuda, M. Minoh, and K. Ikeda. Extracting straight lines by sequential fuzzy clustering. *Pattern Recognition Letters*, 17(6):643–649, 1996.

[89] G. Tzanidou and E.A. Edirisinghe. Automatic baggage detection and classification. In *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pages 825–830, Nov 2011.

[90] G. Tzanidou, I. Zafar, and E.A. Edirisinghe. Carried object detection in videos using color information. *Information Forensics and Security, IEEE Transactions on*, 8(10):1620–1631, Oct 2013.

[91] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *In European Conference on Computer Vision, volume IV*, pages 705–718, 2008.

[92] C.J. Veenman, M.J.T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(1):54–72, 2001.

[93] R. Wang and T. Huang. A framework of joint object tracking and event detection. *Pattern Analysis and Applications*, 7(4), 2004.

[94] Z. Wang, E.E. Kuruoglu, X. Yang, Y. Xu, and S. Yu. Event recognition with time varying hidden markov model. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1761–1764, April 2009.

[95] L. Wolf and S. Bileschi. A critical view of context. *International Journal on Computer Vision*, 69(2):251–261, August 2006.

[96] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

[97] Z. Wu, T.H. Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1185–1192, June 2011.

[98] G. Xu, Y. Ma, H. Zhang, and S. Yang. Motion based event recognition using hmm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 831–834, Aug 2002.

[99] G. Xu, Y. Ma, H. Zhang, and S. Yang. An hmm-based framework for video semantic analysis. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(11):1422–1433, Nov 2005.

[100] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. *CVPR*, 2011.

[101] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[102] Q. Yu and G. Medioni. Multiple-target tracking by spatiotemporal monte carlo markov chain data association. *PAMI*, 31, 2009.

[103] B. Yuan, Q. Ruan, and G. An. Carried object detection in short video sequences. In *Signal Processing (ICSP), 2014 12th International Conference on*, pages 1311–1316, Oct 2014.

[104] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[105] J. Zunic and P. L. Rosin. A convexity measurement for polygons. *PAMI*, 26:173–182, 2002.