HUMAN DETECTION, TRACKING AND SEGMENTATION IN SURVEILLANCE VIDEO

by

GUANG SHU
M.S. Shanghai Jiaotong University, 2009

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2014

Major Professor: Mubarak Shah

ABSTRACT

This dissertation addresses the problem of human detection and tracking in surveillance videos. Even though this is a well-explored topic, many challenges remain when confronted with data from real world situations. These challenges include appearance variation, illumination changes, camera motion, cluttered scenes and occlusion. In this dissertation several novel methods for improving on the current state of human detection and tracking based on learning scene-specific information in video feeds are proposed.

Firstly, we propose a novel method for human detection which employs unsupervised learning and superpixel segmentation. The performance of generic human detectors is usually degraded in unconstrained video environments due to varying lighting conditions, backgrounds and camera viewpoints. To handle this problem, we employ an unsupervised learning framework that improves the detection performance of a generic detector when it is applied to a particular video. In our approach, a generic DPM human detector is employed to collect initial detection examples. These examples are segmented into superpixels and then represented using Bag-of-Words (BoW) framework. The superpixel-based BoW feature encodes useful color features of the scene, which provides additional information. Finally a new scene-specific classifier is trained using the BoW features extracted from the new examples. Compared to previous work, our method learns scene-specific information through superpixel-based features, hence it can avoid many false detections typically obtained by a generic detector. We are able to demonstrate a significant improvement in the performance of the state-of-the-art detector.

Given robust human detection, we propose a robust multiple-human tracking framework using a part-based model. Human detection using part models has become quite popular, yet its extension in tracking has not been fully explored. Single camera-based multiple-person tracking is often hindered by difficulties such as occlusion and changes in appearance. We address such problems by developing an online-learning tracking-by-detection method. Our approach learns part-based person-specific Support Vector Machine (SVM) classifiers which capture articulations of moving human bodies with dynamically changing backgrounds. With the part-based model, our approach is able to handle partial occlusions in both the detection and the tracking stages. In the detection stage, we select the subset of parts which maximizes the probability of detection. This leads to a significant improvement in detection performance in cluttered scenes. In the tracking stage, we dynamically handle occlusions by distributing the score of the learned person classifier among its corresponding parts, which allows us to detect and predict partial occlusions and prevent the performance of the classifiers from being degraded. Extensive experiments using the proposed method on several challenging sequences demonstrate state-of-the-art performance in multiple-people tracking.

Next, in order to obtain precise boundaries of humans, we propose a novel method for multiple human segmentation in videos by incorporating human detection and part-based detection potential into a multi-frame optimization framework. In the first stage, after obtaining the superpixel segmentation for each detection window, we separate superpixels corresponding to a human and background by minimizing an energy function using Conditional Random Field (CRF). We use the part detection potentials from the DPM detector, which provides useful information for human shape. In the second stage, the spatio-temporal constraints of the video is leveraged to build a tracklet-based Gaussian Mixture Model for each person, and the boundaries are smoothed by multi-frame graph optimization. Compared to previous work, our method could automatically segment multiple people in videos with accurate boundaries, and it is robust to camera motion. Experimental results show that our method achieves better segmentation performance than previous methods in terms of segmentation accuracy on several challenging video sequences.

Most of the work in Computer Vision deals with point solution; a specific algorithm for a specific problem. However, putting different algorithms into one real world integrated system is a big challenge. Finally, we introduce an efficient tracking system, NONA, for high-definition surveillance video. We implement the system using a multi-threaded architecture (Intel Threading

Building Blocks (TBB)), which executes video ingestion, tracking, and video output in parallel. To improve tracking accuracy without sacrificing efficiency, we employ several useful techniques. Adaptive Template Scaling is used to handle the scale change due to objects moving towards a camera. Incremental Searching and Local Frame Differencing are used to resolve challenging issues such as scale change, occlusion and cluttered backgrounds. We tested our tracking system on a high-definition video dataset and achieved acceptable tracking accuracy while maintaining real-time performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The goal of computer vision is to let computers understand a scene through the eyes of a camera. With the development of digital camera devices and the rapid growth of video data in recent years, video analysis is becoming one of the most popular areas in computer vision research. At any given moment thousands of videos are uploaded to Internet websites such as YouTube, and each year millions of surveillance cameras around the world are capturing trillions of hours of video. It is impossible for humans to process such large amounts of video data. Therefore computer vision algorithms for human detection, tracking, segmentation, action recognition, video retrieval, abnormal event detection and video summarization are becoming increasingly important.

Object detection and tracking are the most fundamental tasks in computer vision. They have been widely used in many applications such as security and surveillance, human-computer interaction, video communication and compression, augmented reality, traffic control, and medical imaging. Object detection is the process of detecting the instances of a certain class of objects (e.g. humans, dogs, and bicycles) in images and videos. Object tracking is the process of locating moving objects in different frames of a video while maintaining the correct identities.

In the context of video surveillance, object detection and tracking are usually coupled together to locate the objects of interest through the video. We need to first detect pedestrians in each video frame, and then track them across different frames. In this dissertation, we focus on the object class of humans, since humans are most likely to be the object of interest in applications such as visual surveillance, human computer interaction, and autonomous vehicle navigation.

Object segmentation is also an important task in computer vision. Object Segmentation is the process of delineating the target object from the image. Human segmentation can benefit many computer vision applications, such as person recognition, pose estimation, body part tracking, motion analysis, action recognition, and autonomous driving system.

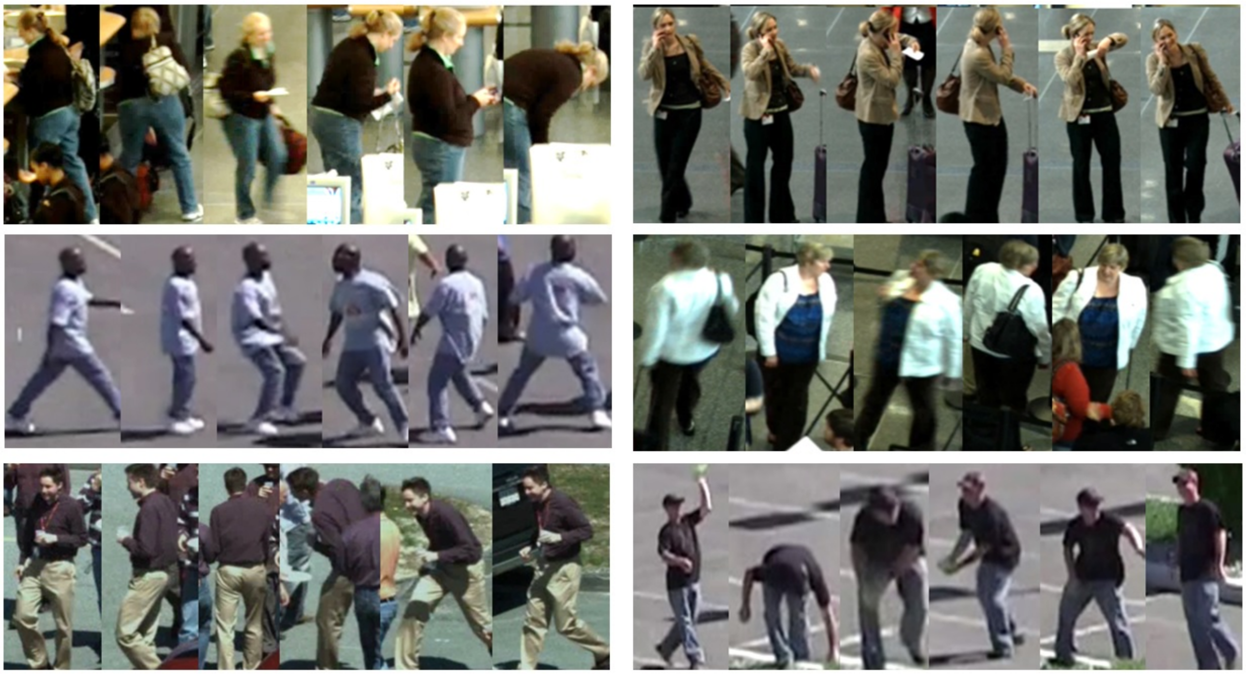


Figure 1.1: Examples of appearance change. Each subfigure shows that a pedestrian may present various appearances in a video, which makes it very difficult to find a universal representation for human detection or tracking.

1.1 Challenges

Object detection, tracking and segmentation have been extensively studied in the past few decades. Although a large number of algorithms have been proposed, problems in the field persist and have detracted from the commercial viability of much of the research.

For the object class of humans, there are additional difficulties in detection and tracking. Firstly, since the human body is able to articulate at many joints, the appearance of humans can vary not only with a changing viewpoint but also with the changing poses and juxtaposition of body parts. Furthermore, people wear a variety of clothes and accessories which, taken together, can constitute thousands of combinations of colors, textures, materials and styles. Thirdly, people tend to carry other objects such as luggage, strollers or bicycles. These factors make it very difficult to find a universal representation for the class of humans. Figure 1.1 shows several examples of pedestrians in a variety of poses in surveillance video.

Another problem is partial occlusion of target objects by other objects. Figure 1.2 shows several examples of partial occlusion in surveillance video. It is difficult for a detector to separate individual people when they are close to each other. A tracking algorithm will incorrectly jump to a nearby object with a similar appearance when the target is partially occluded.

In many cases, partial occlusion is coupled with appearance change or direction change, which makes it even more difficult to track the target. Figure 1.3 shows such an example. The man in the brown coat walks into a crowd. He is then occluded by other people while he is making a U-turn. He then reappears but with a white shirt and his face visible. When tracking such targets, the tracker often drifts to the cluttered background and has difficulty remaining with or reacquiring the target when aspect or details of objects change. The model is not properly updated during the occlusion, and thus cannot reflect the sudden change in appearance or motion.

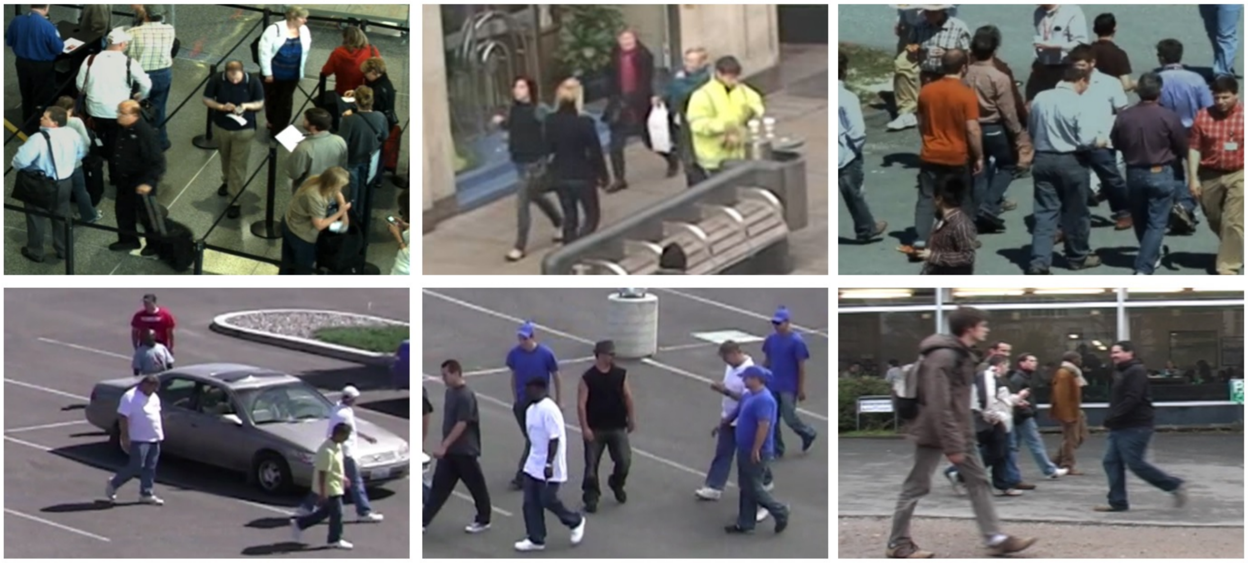


Figure 1.2: Examples of partial occlusion. These partially occluded pedestrians are extremely hard to be detected or tracked.

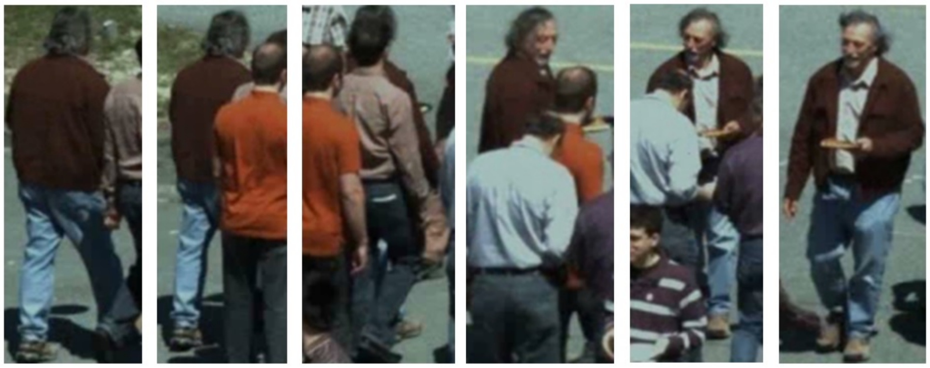


Figure 1.3: An example of occlusion and appearance change coupled together. The man in the brown coat is walking into a crowd. Then he get occluded by other people while he is making a U-turn. Later he reappears with different image appearance.

Early efforts [38, 94–96] in object detection were based on background subtraction. While this approach is quite efficient for detecting isolated moving objects, it is not capable of detecting static objects or separating individual humans when they are close to one another.

Recently human detectors [33, 43, 104] have achieved better performance. However, due to the large variation across training and testing data, a pre-trained model may perform sub-optimally in real world environments.

In object tracking, early approaches [30, 91] suffer in real world environments. A number of recent approaches [9, 11, 28, 35, 49, 54] follow the appearance change, continually updating their model based on a discriminative online-learning paradigm. However, the online detectors still drift over the long-term in the presence of partial occlusions. Several tracking-by-detection methods [13, 21, 56, 111] address the occlusion problem by associating detections over a temporal window, given certain global constraints. However, they still suffer when faced with appearance changes, such as the man revealing a white shirt when turning to face the camera.

Human segmentation from video sequences remains a very challenging task as well. Although general purpose object segmentation has been extensively studied for a few decades, there is still no universal solution to effectively segment multiple human instances in videos. Early image segmentation approaches [4, 23, 73, 80, 86] mostly use edge detection or region splitting techniques. More sophisticated methods are used in recent works: Active contour [55], [17], Mean shift [29, 101], Normalized cuts [90]. Recently, Graph cuts and energy-based methods [18, 19, 89] achieved satisfying results on still images. However, they still cannot handle realistic scenes in a video with multiple objects and complex backgrounds. Also, these methods require relatively good initialization that is usually done manually by a user. This is very expensive for video analysis.

In order for different computer vision algorithms for detection, tracking and segmentation to be useful in real world, they have to be implemented in a real time system. A computer vision system should use simple, reasonable, but reliable and fast computer vision algorithms to meet the real world requirements. However, many sophisticated computer vision algorithms cannot achieve real-time performance, especially as the growth in image resolution. For example, the fast implementation of DPM detector [76] runs at 1 frame/second for a VGA (640×480) video, but only 0.15 frame/second for a HD (1920×1080) video. On the other hand, many existing computer vision systems are not capable of handling the more complex tasks nowadays. Previous video surveillance systems such as Cocoa [6, 15] and Knight [75] are designed to work only in simple scenes with a few objects. They are not able to track people in highly crowded scenes such as an airport terminal. Therefore, a real-time video surveillance system that is capable of handling high-resolution video and complex environments is highly desirable.

1.2 Proposed Works and Contributions

To overcome the aforementioned problems, this dissertation proposes several new methods for the detection and tracking of humans in real-world surveillance video. More specifically, we propose new representations for humans and learn important scene-specific information from a video itself to optimize performance of the algorithms in each video. The scene information, such as color, spatio-temporal structure, video resolution, context and background are implicitly encoded into the proposed representations. The key idea behind our approach is to leverage scene-specific information to optimize the performance for various test environments. The experiments show that the proposed methods significantly improve performance in the chosen tasks.

In this section, we introduce methods for three video surveillance tasks. Firstly, we introduce a human detection method which improves a general human detector when it applies to a video. This method uses an unsupervised learning framework to train a specific classifier for the video. Next we describe a robust tracking-by-detection method for semi-crowded scenes. We use a part-based model to handle the appearance changes and partial occlusions in order to improve the overall tracking performance. Thirdly, we introduce a novel method for multiple human segmentation in videos. This method utilizes the part-based detection and the spatio-temporal information of the video. Finally, we introduce a practical, real-time tracking system for single-target tracking in high-resolution video.

The overall contributions of this dissertation are summarized below:

- We propose an unsupervised learning framework for improving human detectors in videos. We also propose a novel superpixel-based model that learns the appearances of humans in a given video. The superpixel-based model learns scene color features, which is an important complement to the HOG features used by a general human detector. Experiments show that the proposed method improves performance of a DPM human detector by up to 15%. This method is fast and easy to implement, therefore it can be used for quickly improving human detection in practical applications.

- We propose a new part-based multiple-human tracking framework that learns in an online fashion a discriminative model for each pedestrian. We use the part-based model to address the tracking in presence of appearance changes and partial occlusion. We also extend a DPM algorithm for detecting partially occluded humans at the detection stages, which significantly improves detection of partially occluded humans. Experiments show that the proposed method improves tracking performance in crowded scenes.

- We proposed a novel multiple human segmentation method. Compared to previous object segmentation methods that need manually interaction, our method can segment multiple humans in videos automatically. We use the part-based detection potentials for reliable initialization; we also leverage the spatio-temporal information to smooth the segmentation boundaries. As demonstrated by comprehensive experimental results, our method is superior over previous human segmentation methods.
- We develop an efficient single-target tracking system for high-resolution video. This tracking system employs fast, easy-to-compute features and multi-threaded architecture to achieve real-time performance. We also use several efficient techniques such as local frame differencing and incremental searching to maintain an acceptable tracking accuracy.
- We have altogether developed two datasets (PNNL Parking lot and Airport) for high-resolution video surveillance research.

1.2.1 Human detection using superpixel segmentation

The first algorithm developed in this dissertation improves the performance of a generic human detector in unconstrained environments. Traditionally, a detector is trained using a limited number of available datasets, therefore it may perform sub-optimally in unknown test environments. Moreover, it is expensive to manually collect new training data from all the test environments. To address this problem, we propose the use of an unsupervised learning framework to iteratively refine the output of a human detector. More specifically, we use a trained DPM detector to collect confident examples from the video itself, and then learn a new classifier with these new examples. Since these examples reflect scene-specific information such as camera angles, resolutions, lighting conditions and backgrounds, the new learned classifier has more discriminative power. Furthermore, this unsupervised learning is performed in an iterative fashion: in each round we collect limited new examples using the classifier learned in the previous round. In this way, we update the classifier conservatively and avoid most of the noise from the original detector.

To train an effective classifier for detecting humans, we introduce a new representation based on superpixels. In particular, each pedestrian is represented using a superpixel-based Bag-of-Words (BoW) model. BoW [64] is a model which represents the occurrence of a vocabulary of image features. It has been widely used in document classification, image classification and action recognition. BoW allows us to build a statistical model over the superpixels which is invariant to pose changes. A superpixel is an aggregation of a number of neighboring pixels that share similar properties such as RGB color. As a middle-level feature, superpixels enable us to measure the feature statistics on a semantically meaningful sub-region rather than individual pixels which can be brittle. On the other hand, the superpixels have great flexibility which avoids the misalignment of the histogram of gradient (HOG) and Haar-like features on variant poses of objects.

There are two advantages of using the color-based representation. Firstly, in a typical surveillance video, each individual usually moves from one side of the scene to the other, therefore typically many instances in the spatio-temporal domain are captured. These instances have variable poses but consistent color features. Second, color features are complementary to HOG features used in a DPM detector, therefore provide additional information.

Our approach for improving a human detector consists of three main phases. First we apply a DPM detector with a low detection threshold to every frame of a video and obtain a substantial number of detection examples. These examples are initially labeled as positive, negative or hard based on the confidence levels of the DPM detector. Secondly, we segment detection windows and obtain superpixels to make a BoW representation for each example. In the last step, we train an SVM model with positive and negative examples and label the hard examples iteratively. Each time a small number of hard examples are conservatively added into the training set until the iterations converge.

In summary, the value of this work is in the following areas: Firstly, we are able to adapt a generic detector to a specific video using an unsupervised learning framework. Secondly, we have developed a superpixel-based BoW representation which is effective for human detection. Thirdly, our method is very efficient and can be used for quickly improving human detection in practical applications.

1.2.2 Robust Multiple-human Tracking with Occlusion Handling

The second algorithm developed in this dissertation addresses multiple human tracking in a video of a semi-crowded environment obtained by a single camera. Traditional tracking methods suffer from the problems of appearance changes and occlusion. To resolve these problems, we follow a tracking-by-detection framework in which we learn a person-specific classifier in an online fashion to capture the appearance changes. More importantly, we propose to employ a part-based human model as an effective representation for tracking humans.

There are several advantages of the part-based representation. The combination of parts provides a rich description of the articulated body, thus it represents the human form better than a whole detection window. Second, the part-based model excludes most of the background from within the detection window and thus avoids confusion from background changes. Finally, since the part-based detector is trained using a large number of human examples, it captures a significant amount of discriminative information, which is essential for tracking.

The part-based representation also allows us to handle partial occlusions in crowded scenes. Since the combination of parts is very flexible, a person with partial occlusion may still have a few body parts visible. In the detection stage, we select the subset of parts which maximizes the probability of detection. This significantly improves detection performance. In the tracking stage, we form judgments about the partial occlusion of a person using the person-specific classifier. This inferred occlusion information is used to update the classifier, which prevents performance from suffering during the occlusion period. Also, the discovered occlusion information is passed to the next frame in order to penalize the contribution to the scoring algorithms of the occluded parts when applying the person classifier.

Our multiple-human tracking algorithm is composed of four steps. First, we use an extended part-based human detector on every frame and extract the part features from all detections. Next, person-specific SVM classifiers are trained using the detections, and consequently used to classify the new detections. Then we use a greedy bipartite algorithm to associate the detections with the trajectories, where the association is evaluated using three affinity terms: position, size, and the score of the person-specific classifier. Finally, trackers are updated using occlusion reasoning.

In summary, this makes the following contributions: Firstly, we adopt the part-based model in an online-learning framework to tackle occlusion and appearance changes. Secondly, we extend the DPM human detection algorithm which allows us to improve the detection in crowded scenes. Thirdly, we propose a dynamic occlusion handling method to learn and predict partial occlusions, and consequently improve the tracking performance.

1.2.3 Multiple Human Segmentation based on Detection and Multi-frame Optimization

The third algorithm developed in this dissertation addresses multiple human segmentation in a video. Traditionally, human detection and segmentation are solved using different methodologies. However, human segmentation can benefit considerably from a human detector, which can be used as an automated initialization for segmentation. The deformable part-based model (DPM) human detector [44], which is state-of-the-art, has achieved acceptable detection performance. It localizes multiple humans in a video and provides bounding boxes, which can be used to initialize the segmentation. The detection output can prune most of the background area in the video, thereby reducing the computational cost and improving the accuracy of human segmentation in the video. Moreover, the part-based detector can detect the essential shape structure of the human, which provides a good estimate of the foreground confidence density and bias the segmentation.

On the other hand, human detection can also benefit from segmentation. Although a human detector can localize objects in a video, the bounding boxes are usually not precise enough for many other applications. Detector-based segmentation can remove background distractions and provide clean object regions instead of the rough original output. Anelia [8] shows that object segmentation can significantly improve the object classification performance instead of using detection outputs.

In this dissertation, we propose a two-stage coarse-to-fine human segmentation approach. In the first stage, we employ the part-based human detector and obtain both human detections and many background examples, then we perform a superpixel-level segmentation for each detection window. We formulate the segmentation problem as binary labeling and solve it using Conditional Random Field (CRF) optimization. In order to get the background probability distribution, a Gaussian Mixture Model (GMM) is built using superpixel-level features extracted from all the background examples of the entire video sequence. With the background GMM model, we can obtain the background probability for each superpixel in the human detections.

The advantages of using superpixels for initial segmentation are three fold: first, a typical human detection window usually consists of over thousands of pixels but less than a hundred superpixels, therefore it is more efficient to calculate. Second, superpixels are low-level segments by color, therefore they naturally preserve the boundary of objects rather than individual pixels, and provide smoother boundaries. Third, the superpixels can represent long-range spatial relations compared to pixels, and this property allows us to build graph-based models at the higher level.

Although color is the primary discriminative feature for segmentation, it is not able to differentiate humans and background when they present the same color, e.g., For example, a person wearing gray is difficult to segment from a gray road. Fortunately, the part-based detector provides approximated part positions based on gradient features; the part structure, including head, shoulders, torso, legs and arms, can be used to estimate the distribution of a person.

Instead of using color distribution as foreground probability, we use the part-based detection potential which is obtained from the responses of the part filters of the detector. Since it is based on HOG features, the detection potential provides a more accurate description of the human body distribution. Different from [65, 71] that use a set of generative shape priors, the detection potential is specific to each human instance in the video therefore providing a more precise description of the unique human shape. With the detection potential and the color distribution, each detection window is segmented using CRF optimization.

Although reasonable, the initial segmentations still have noise caused by shadows or occlusions that cannot be handled at superpixel-level. Therefore we need a pixel-level segmentation to refine the object boundaries. Given the initial segmentations, we build a Gaussian Mixture Model for each individual. Instead of using only one detection window, we built a multi-frame GMM based on each tracklet that consists of a few consistent detections in the spatio-temporal domain for the same person. The tracklet provides more appearance constraints which helps reduce noise. Finally, a multi-frame CRF optimization is applied on each tracklet to obtain the final segmentations at the pixel-level.

The main contributions of our method are: Firstly, we use the part-based detection potentials of the DPM detector for reliable initialization; shape information is leveraged to improve the performance. Secondly, we leverage the spatio-temporal information in the video by building tracklet-based GMMs and using multi-frame graph optimization, therefore improving the segmentation performance. Thirdly, compared to previous segmentation methods which focus on a primary object or needs manual initialization, our method can segment multiple humans in videos automatically. It requires no manual interaction, is not sensitive to camera motion, and achieves better performance.

1.2.4 NONA: an efficient tracking system for high-definition videos

High-definition surveillance cameras are increasingly being deployed for surveillance nowadays, and the video's resolution can be a hundred million pixels or more. For example, The Image

System for Immersive Surveillance (ISIS), provides a 100 mega-pixel, 360 degree view of the coverage area. MIT Lincoln lab is developing the second-generation system which will ultimately provide a 600 mega-pixel field of view with a refresh rate of once per second. Although the high-resolution video provides much better image quality and thus better facilitates video surveillance tasks, data throughput and computational expense considerations mean that most current tracking methods cannot achieve real-time performance with these multi-megapixel video streams.

We have developed an efficient tracking system with real-time performance for high-resolution, real world surveillance video. The NONA tracking system is as part of the Wild Area Surveillance project for object tracking. The key functionality is to be able to track a selected target in a crowd in high-resolution video. The major design decisions in NONA include the following:

- Use of multi-threading to enhance system performance. Since video input and display takes the majority of time we employ a multi-threaded system to simultaneously process video frame input, object tracking and frame display, which almost doubles the processing speed over a single-threaded system.
- Use of fast computing features and the Fast Fourier transform (FFT) correlation algorithm to reduce computational cost.
- Use of OpenCV library, which provides a much faster implementation than MATLAB.

While our system achieves real-time performance, the tracking accuracy is not compromised. What is more, we have developed several efficient methods to improve the tracking accuracy. The most common problems in our surveillance data are occlusions and cluttered backgrounds. These problems cause the tracker to quickly drift into the background where the features are similar to the target when the target is temporarily occluded. They remain with the background even when the target reappears. To resolve this problem, we propose a local frame differencing technique, which can detect foreground and background in the local search region. This technique is very efficient to calculate and very effective in keeping the tracker from drifting into the background. It significantly improves the overall tracking accuracy.

Another technique involved in the system is adaptive template scaling, which is used to handle the problem of scale changes. Since the cameras are set with various angles from 30 degrees to directly below, the size of the target varies at different distances due to the perspective of the scene, which causes unstable matching. We use multi-scale templates in the template matching to help the tracker follow the scale changes of the target.

In summary, this work makes the following contribution: The NONA tracking system can achieve real-time performance on all kinds of video sequences with fast algorithms and a multi-threaded implementation. Several efficient and effective techniques are used to improve the tracking accuracy. In addition, it remains a general tracking system. Since it does not require any prior knowledge of the target or scenario, it can track not only pedestrians, but also vehicles, baggage or animals in all kinds of video sequences.

1.3 Organization of the Dissertation

The dissertation is structured as follows:

Chapter 2 reviews existing literature on object detection and tracking. Chapter 3 proposes a semi-supervised learning framework for improving a generic detector in unconstrained video. Chapter 4 presents a part-based online-learning tracking framework to track multiple humans in surveillance video. Chapter 5 introduces a novel method for segmenting multiple human in videos. Chapter 6 introduces a real-time tracking system for tracking in high-resolution video data. Chapter 7 describes a summary of contributions and a description of future work.

CHAPTER 2: LITERATURE REVIEW

Object detection, tracking and segmentation are three fundamental tasks in computer vision. In this chapter, we review a number of relevant works in literature related to the two tasks. We present prominent works in object detection including several of the most popular object detectors, together with a few recent methods that leverage unsupervised learning for object detection. We also present some object tracking work in two classes; single object tracking and multi-object tracking. In addition, we present object segmentation work including early approaches and recent graph optimization approaches. We discuss their respective advantages and drawbacks and describe where our work fits into the context of these methods.

2.1 Object Detection

The goal of object detection is to detect and localize a class of object in image or video. Three most studied object classes are Face [68, 99], Pedestrian and car [58]. In this dissertation we focus our study on pedestrian detection, but our proposed methods can also be extended to other classes with minor modifications.

Early works [69, 98] detect pedestrians using Haar-like features which have been successfully applied in face detection [99]. The Haar-like features are rectangular features similar to Haar basis functions which have been used by Papageorgiou et al. [77]; These simple features can be computed very rapidly using an the integral image.

Viola et al. in [98] trained a human detector using an Adaboost algorithm, which selects a small number of classifiers iteratively from a large set of possible weak classifiers. The selected weak classifiers are combined into an effective classifier for detection. Mohan et al. [69] present a detection framework that uses four example-based detectors to localize the head, legs, and arms on human body. Although these methods are very efficient, the Haar-like features are not capable of handling the complexity in real-world images and videos.

Background subtraction [38, 94–96] is also an efficient way to detect moving objects. It extracts the foreground regions by calculating the difference between a video frame and the background model. However, it has a few limitations: it requires the video is taken by a static camera - only moving objects will be detected; it is difficult to separate humans in a group; and since it does not have a specific appearance model for the target, it cannot differentiate humans from other moving objects, for example cars or animals.

In 2005, Dalal and Triggs [33] proposed a Histogram of Oriental Gradients descriptor (HOG) for object detection. In their method, a set of overlapping HOG features are extracted from a sliding window and then fed into an SVM [31]. The advantage of HOG features is that each image cell is statistically represented by a histogram of the gradient orientations and magnitudes, thus it is more invariant to illumination, shadows, etc.

HOG features have been widely used in pedestrian detection and a number of improved methods have been proposed based on HOG features. Wang and Han [103] proposed a HOG based detection framework that combines HOG features and Local Binary Pattern (LBP) [5] feature, which captures the subtle intensity changes in a small local area. Due to its superior performance and computational efficiency, LBP has become a popular feature for texture classification, face recognition, and object detection. More importantly, they presented a partial occlusion handling method for still images which improves the detectors performance in cluttered scenes. In particular, they construct an occlusion confidence map by clustering the responses of blocks in a detection window.

Walk et al. [100] introduced a new color self-similarity (CSS) feature which learns the color patterns in humans. This feature captures pair-wise information of spatial color distributions, thus being a useful complement to the HOG feature.

Dollar et al. [36] proposed a fast detection method that runs in real-time with almost the same performance as HOG detector. Their recent survey [37] has summarized the state-of-the-art pedestrian detectors.

Part-based model has been explored in early works [46, 67, 106]. In 2008, Felzenszwalb, McAllester, and Ramanan [44] proposed an object detection algorithm based on deformable part-based model. This model is defined by a root filter, several part filters, and a deformable model which weights the configurations of the parts. All the parts are trained on HOG features at multiple scales using latent SVM. Furthermore, this approach uses mixture models to deal with intra-class variations, e.g. frontal versus side views. A DPM system won the 2007 PASCAL challenge and is still the state-of-the-art in object detection.

The above-mentioned methods are designed for image-based object detection, however, when video is available, additional information such as optical flow or motion constraints can greatly assist the detection task. Viola et al. [98] extracted Haar-like features from optical flow and combined it with appearance features. Dalal et al. [34] proposed a new histogram of optical flow feature and combined it with HOG feature. However, the computational cost of calculating motion features is relatively high.

More recently, detection-by-tracking approaches [7, 14, 57, 105] use motion constraints in the video to optimize tracking and detection simultaneously. Andriluka et al. [7] combined human detection and human tracking in a single framework, and proposed a pictorial structures model. In [14, 57, 105], object hypotheses are detected in all frames and then associated by trackers. By using the structural information of the scenario, the tracker may find even the occluded objects. However, tracking itself is a challenging problem. If the tracker is not reliable in the scene, it may lower the performance of the original detector.

2.2 Unsupervised Learning for Object Detection

Due to the large variation between training and test data, an object detector trained using a limited number of available datasets may perform sub-optimally on unknown test data. Moreover, it is expensive to manually collect new training data from different test environments.

Unsupervised learning is an effective way to resolve this problem by learning the scene-specific information in a video. There has been a substantial amount of work that addresses the problem of learning from unlabeled data in a semi-supervised fashion [27,61,72,74,87,104,109]. A common technique of these approaches is to apply a coarse detector to the video and get initial detections, which are then added into the training set to improve the coarse detector. Levin et.al [61] built a semi-supervised learning system using co-training, in which two different classifiers are used to train each other to improve the detection performance. In [74] a co-training based approach is proposed to continuously label incoming data and use it for online updates of the boosted classifier. However, both approaches require a number of manually labeled examples as the initial training examples. Authors in [72] presented a framework that can automatically label data and learns the classifier for detecting moving objects from video. Celik et.al [27] proposed an approach to automatically detect dominant objects for visual surveillance. However, in [72] and [27] the initial coarse detectors are based on background subtraction, hence they will not work that well in the scenarios with complex background or moving camera. Wang et.al [104] proposed a non-parametric transfer learning approach, in which they use a vocabulary tree to encode each example into binary codes. While these approaches have proven effective, they can only adapt their appearance models based on the coarse detections and so are not truly adaptive, e.g. Wang et al. [104] only learns objects having the similar appearance to the initial examples. These approaches are likely to miss some hard examples with large variations in appearance.

Our human detection approach is based on an unsupervised learning framework. Compared to previous methods, the major difference is that we train a second classifier with a superpixel-based BoW feature. There are three advantages of this feature. Firstly, a superpixel is an aggregation of a number of neighboring pixels that share similar properties such as RGB color. Superpixels make a more semantically meaningful representation for humans than pixel-grids. Secondly, encoded with color information, superpixel-based features are an important complement to gradient features. Thirdly, BoW allows us to build a statistical model over the superpixels which is invariant to pose changes.

2.3 Single Object Tracking

Object tracking is one the most fundamental problems in computer vision. Yilmaz et al. [108] summarized most of the single object tracking methods. Early works [24, 91] focus on estimate the motion parameters of the target. Shi et al. [91] uses the affine motion model to compensate the image motion and selects discriminative feature points for the tracker. Comaniciu et al. [30] proposed the mean-shift algorithm to track objects by finding the peak in a confidence map of the surrounding area. This confidence map is a probability density function calculated on each pixel using color similarity. The above-mentioned works provide basic tracking frameworks. However, the object models in these methods are too simple to handle real world videos.

Later discriminative tracking approaches [9, 11, 28, 35, 49, 54] with online-learning have been extensively explored. Avidan [9] train an ensemble of weak classifiers using AdaBoost to distinguish between the object and the background. Collins and Liu [28] selects the best features to track based on the two-class variance ratio between foreground and background. Grabner et al. [49] learns a discriminative classifier in an online manner to detect the object from the background. Babenko et al. [11] use Multiple Instance Learning (MIL) to avoid the drift caused by inaccurate bounding boxes.

In such methods, an object-specific detector is trained in a semi-supervised fashion and then used to locate the target in consecutive frames. In particular, a few positive and negative examples are collected each time the target is located, and these examples are immediately used to re-train the detector. In this way, the detector can capture the most discriminative features on the object. However, the online learned detector will often drift into the background in long-term tracking.

In recent years, a few sophisticated methods for single-target tracking have been proposed to address complex real world data. Kala [54] leverages the video structure to select the most confusing negative examples from backgrounds to strengthen the classifier. Dinh et al. [35] tracks multiple objects beside the target, therefore it avoids the confusions with other objects with similar appearances. Hare et al. [51] present a framework for adaptive visual object tracking based on structured support vector machine, which provides superior tracking performance.

With sophisticated models and online-learning approaches, these methods are more invariant to appearance change and occlusion. However, their computational cost is relatively high for a practical tracking system. More importantly, single object tracking methods are not able to handle multiple people in real world video surveillance data, as shown in 2.1.

2.4 Multiple Object Tracking

A large number of works [13,16,21,56,109] have covered multi-target tracking-by-detection algorithms. These methods tackle multi-target tracking by optimizing detection assignments over a temporal window, given certain global constraints. Blackman [16] proposed a multiple hypothesis tracking framework; possible trajectory hypotheses are proposed and propagated into the future in anticipation that subsequent data will solve the difficult data association decisions. Zhang et al. [109] resolve the association between the detection and the tracking by optimizing a cost-flow network with a non-overlap constraint on trajectories. Brendel et al. [21] apply a maximum-weight independent set algorithm to merge small tracklets into long tracks. Benfold et al. [13] use Markov-Chain Monte-Carlo Data Association (MCMCDA) to correspond the detections obtained by a HOG-based head detector in crowded scenes. Such methods employ offline-trained detectors to find the targets and associate them with the tracks.

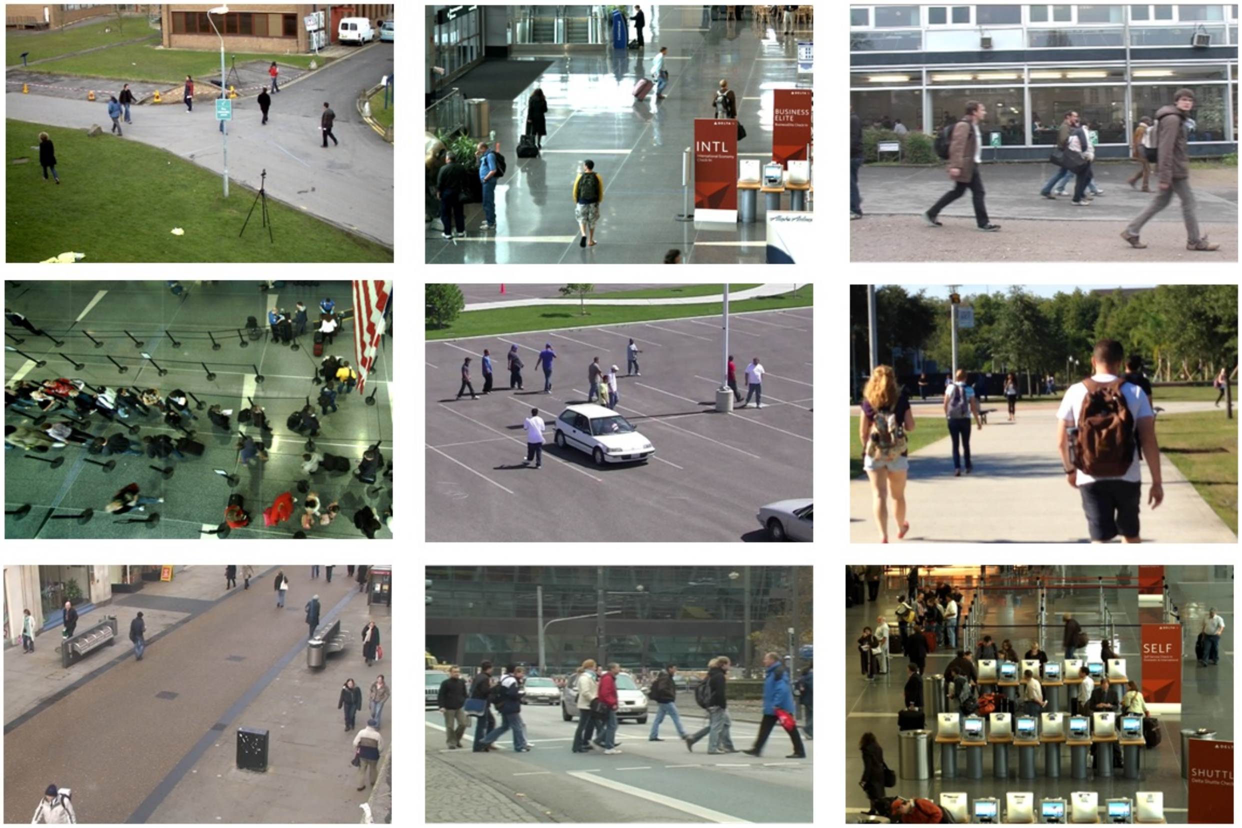


Figure 2.1: Screenshots from real world tracking datasets.

Although they can handle several difficulties, such as the uncertainty in the number of targets, occasional occlusions, and template drift in long term; they still suffer when faced with appearance changes and occlusion. In particular, when tracking a crowd of pedestrians, the data associations often fail in the aforementioned approaches due to pose variations, partial occlusions and background changes. Furthermore, these methods employ the information from future frames to locate the targets in the current frame with a temporal delay, which is not always available for a practical surveillance system. In contrast, we employ a greedy scheme in data association, which is more suitable for real-time tracking applications.

On the other hand, several methods such as [59,81,107] employ social force models which consider the interactions between individuals in order to improve the motion model. Such methods require prior knowledge of the 3D scene layout, which is often unavailable in real world scenarios.

The method proposed by Breitenstein et al. [20]is evidently the most similar to our proposed multiple human tracking method. In their paper they propose a particle-based framework in which detections and intermediate detection confidences are used to propagate the particles. Additionally, they employ a target-specific classifier to associate the detections with the trackers. Our method is different from [20] in that we employ a part-based appearance model which is more robust and can handle partial occlusions.

2.5 Human Segmentation

A common approach for human segmentation in videos is to use background subtraction [38, 94–96]. Background subtraction can extract the foreground silhouettes of objects by calculating the difference between a video frame and the background model. However, there are a few limits to background subtraction; first, it requires that the video is taken by a static camera; second, only moving objects will be detected; third, it is difficult to separate humans in a group.

Graph cuts and energy-based methods [18, 19, 89] have achieved satisfying results on still images. There are some recent works extending the graph cut methods to video object segmentation by considering motion and appearance constraints in a video. Price [83] and Bai [12]achieved satisfying segmentation results but they require manually initialization by a user. In video surveillance there is a huge amount of video data, therefore manual initialization is not practical. Consequently, automated methods [26, 50, 52] have been proposed. Brendel [22] and Brox [25] also utilize an object's trajectory and motion cues. However, these methods do not have an explicit model of the object and so the segments usually do not correspond to an actual object, only to image regions with similar appearance or motion. To solve this issue, object proposal-based methods [60, 66, 110] are proposed for video object segmentation. The limit of these methods is that they can only detect and segment the primary object in the video. They are not capable of handling a typical surveillance video with multiple humans in a single frame. Also they optimize over a long period, which is high on computational cost.

Superpixels have been successfully applied in image segmentation [4]. As the middle-level feature, it provides a more natural representation than pixel-grids. It enables us to measure the feature statistics on a perceptually meaningful sub-region that has consistent color or texture. It has great flexibility for non-rigid objects, which avoids the mis-alignment of the HOG and Haar-like features on variant human poses. The pairwise constraints between superpixels can represent long-range spatial relations compared with pixel-grids, and this property allows us to build graph-based models at the higher level. Most importantly, it is computationally efficient: It reduces the complexity of an image from hundreds of thousands of pixels to only a few hundred superpixels.

Our work lies between human detection and object segmentation. We focus on leveraging the detector's ability to segment multiple humans in a video. The output of our method can be seen as an improvement to human detection. Instead of bounding boxes outputted by the detector, our method obtains a more precise human silhouette, which serves as a more precise initialization for other applications such as person recognition, pose estimation, action recognition, and human tracking. Recent works [8, 32, 78, 79] demonstrated that object segmentation can improve performance significantly in recognition tasks. Therefore, human segmentation is a very promising area which will be very important to the development of other applications.

CHAPTER 3: HUMAN DETECTION USING SUPERPIXEL SEGMENTATION

Object detection is one of the most important tasks in computer vision. It has many applications including face detection, object localization and autonomous car driving. It also serves as a necessary pre-processing step for many other applications such as face recognition, object tracking, activity recognition and scene understanding. The demand of automatically detecting objects in videos has significantly increased with the rapid development of video recording devices in recent years.

Humans are one of the most difficult object classes to detect because the articulated bodies present various appearances. The state-of-the-art detectors based on HOG features are not able to encode all the variations in the model. In addition, the rich color cues are typically not leveraged by human detectors since the color of clothes is not constant.

In this chapter, we introduce a method that improves the performance of a generic human detector when applied to a specific video. The contributions of our work are twofold. Firstly, we propose an unsupervised learning framework which learns scene specific information. In particular, our method automatically discovers new human and background examples which serve as more effective training data for the specific video. Secondly, we propose a novel superpixel-based appearance model to implicitly encode the rich color cues, which is crucial to distinguish humans from backgrounds.

3.1 Motivation

The state-of-art human detector, DPM [43], has achieved acceptable performance levels when detecting humans in images. However, its performance in videos is limited for two main reasons: Firstly, it is trained off-line using a fixed set of training examples, which cannot cover all the unconstrained video environments with variable illumination, backgrounds and camera viewpoints. It is also very expensive to manually label examples and re-train the detector for each new video. Secondly, it is designed for a generic object class using HOG [33] features. When applied to a particular video, they are not able to leverage the scene specific information presented in different frames of the video such as the consistent color pattern of objects and the background.

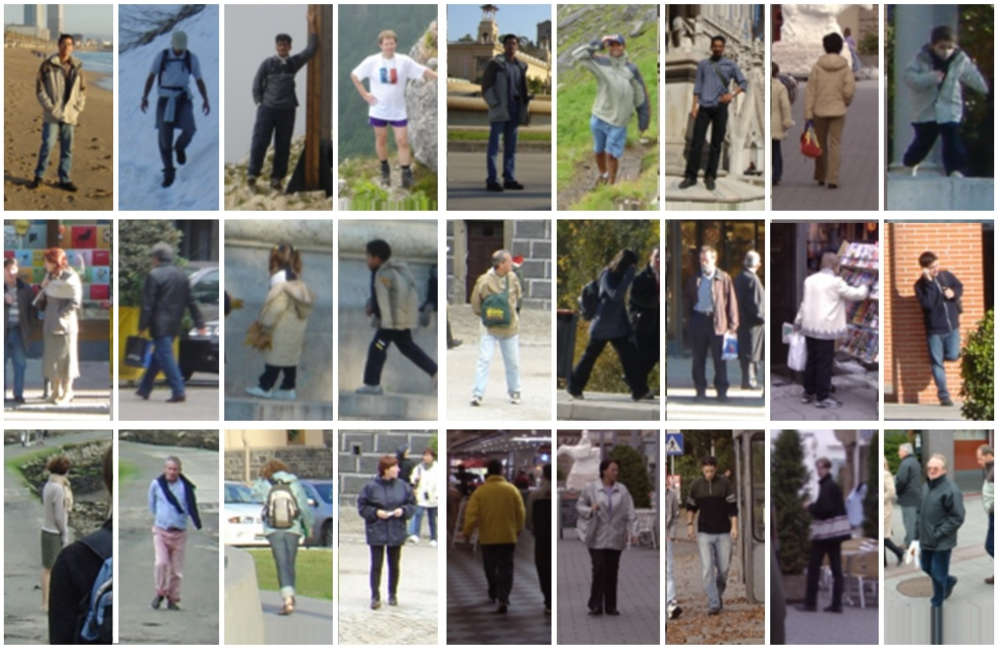


Figure 3.1: A few examples in the INRIA Person Dataset, which is used to train the DPM human detector. Most humans in this dataset are in standing positions with either front view or profile view

A DPM human detector is offline trained with datasets where most humans are in standing position. Figure 3.1 shows the INRIA person dataset, which is a widely used dataset for training DPM. However, in real world video data, humans may change pose to a greater or lesser degree, whereupon the DPM detector can fail. Figure3.2 shows one video frame in which the skateboarder has an unusual pose, and the DPM detector fails to detect the image as a person. Instead, it detects a few false postives in the background, where the HOG features resemble humans.

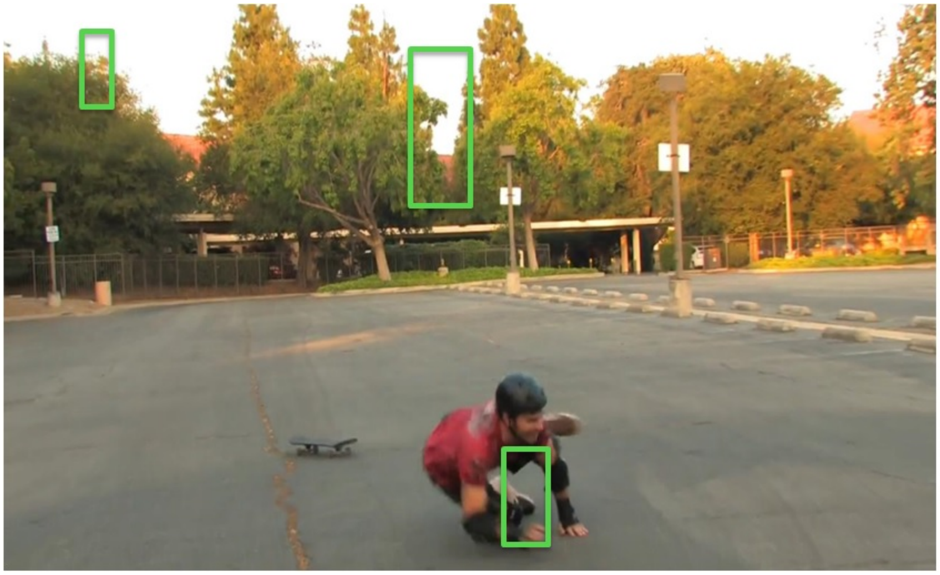


Figure 3.2: DPM detection results in one frame of a skateboarding video. The green bounding-boxes show the output of DPM detector. The boarder is not correctly detected because the pose (falling over) is quite different from the standing pose. The DPM detector cannot handle the large variation. There are also some false positives in the background because the corresponding HOG features resemble humans.

To address these problems, we propose the use of an unsupervised learning framework to iteratively refine an offline-trained detector for a specific scene. In particular, we introduce a superpixel-based appearance model which encodes the scene-specific color information. The intuition is that in a typical surveillance video, each individual usually moves from one side of the scene to the other side, therefore typically many instances in the spatio-temporal domain are captured. These instances have variant poses but consistent color features. Based on this assumption, we transfer the knowledge from the generic detector to a scene-specific detector by using a novel superpixel-based appearance model.

The rest of the chapter is organized as follows: In section 3.2 we describe the unsupervised learning framework, which includes initial detection (3.2.1) and Classification (3.2.2). In section 3.3 we introduce the superpixel-based model. Experimental results are presented in Section 3.4.

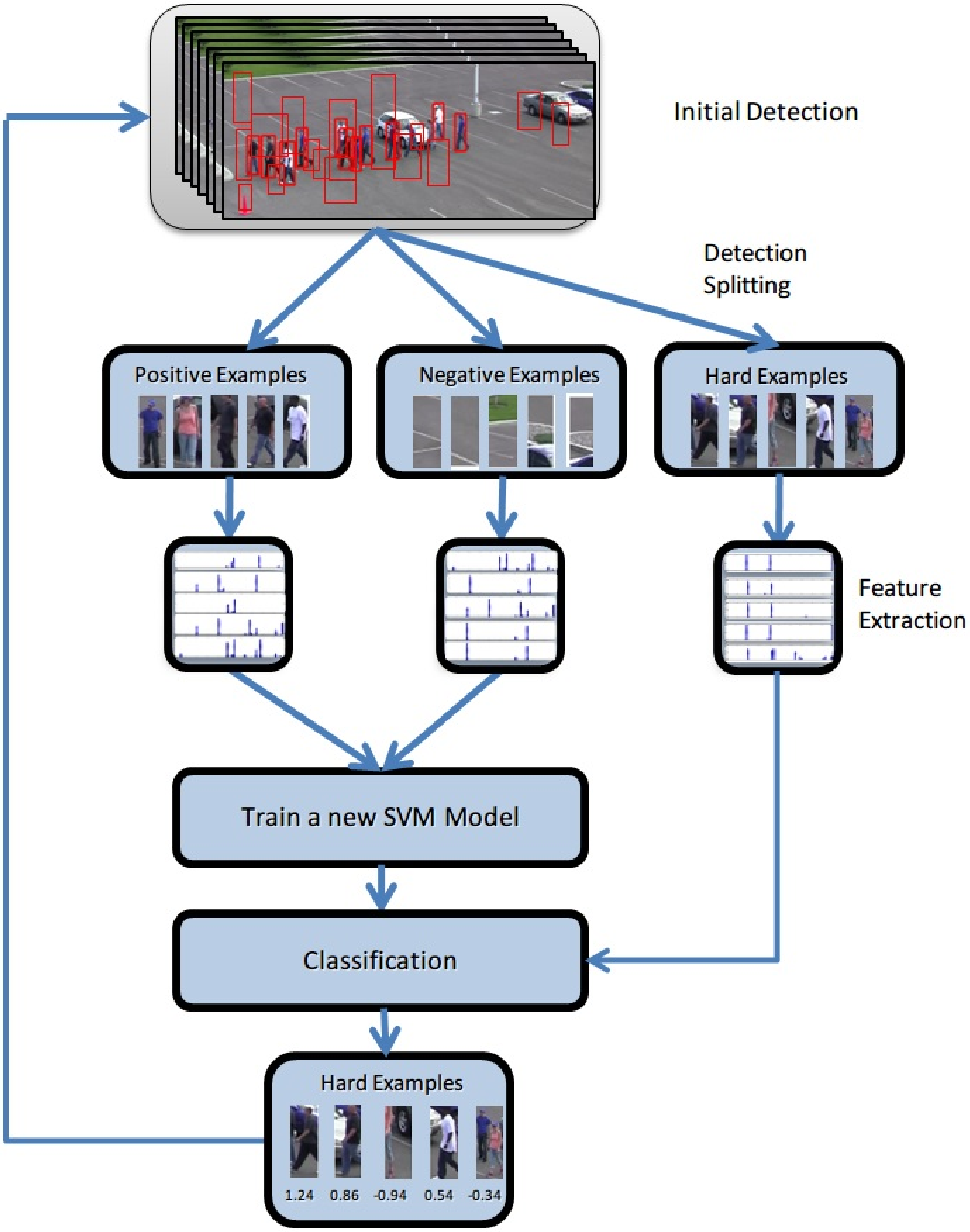


Figure 3.3: The flowchart of our approach for improving a DPM detector.

3.2 Unsupervised Learning Framework

The overview of our approach is illustrated in Figure 3.3. First we apply the generic DPM human detector with a low detection threshold on every frame of a video and obtain a substantial number of detections. Those examples are initially labeled as positive, negative or hard based on their confidence scores. Second, we extract features by segmenting the detection windows into superpixels and making a BoW representation for each detection. In the last step, we train a new SVM classifier with the positive and negative examples and then re-label the hard examples iteratively. In each iteration a small number of hard examples are added into the training set until the iterations converge.

3.2.1 Initial Detection

We employ the deformable part-based model detector [43] as the initial detector in our approach since it has shown excellent performance in static images. The DPM detector is trained using INRIA Person Dataset [33], which consists of hundreds of manually annotated human examples. We applied the DPM detector to a video and collect a large number of initial detection examples $\mathbf{D} = \{d_i \mid i = 1, 2, ..., N\}$. The detector is given a lower detection threshold (high recall) so we can obtain almost all true detections and a large amount of false alarms.

The detector returns a confidence score associated with each $d_i \in \mathbf{D}$, which we denote as $c(d_i)$. According to the confidence scores, we initially split $\mathbf{D}$ into three sets $\langle \mathbf{P}, \mathbf{N}, \mathbf{H} \rangle$, where

$$\mathbf{P} = \{d_i \mid d_i \in \mathbf{D}, c(d_i) > t_p\}, \tag{3.1}$$

$$\mathbf{N} = \{d_i \mid d_i \in \mathbf{D}, c(d_i) < t_n\}, \tag{3.2}$$

$$\mathbf{H} = \{d_i \mid d_i \in \mathbf{D}, t_n \leq c(d_i) \leq t_p\}. \tag{3.3}$$

Figure 3.4 shows the detection splitting result: the ones with detector confidence scores above a threshold t_p are labeled as the positives $\mathbf{P}$; the ones with confidence scores below a threshold t_n are labeled as the negatives $\mathbf{N}$; and the rest are temporarily labeled as hard examples $\mathbf{H}$ whose labels are to be determined. The hard examples usually contain both negatives and hard-to-detect positives with partial occlusions or large variations in pose. In addition, more negative examples are randomly collected in a way that they do not overlap with any positive or hard examples.

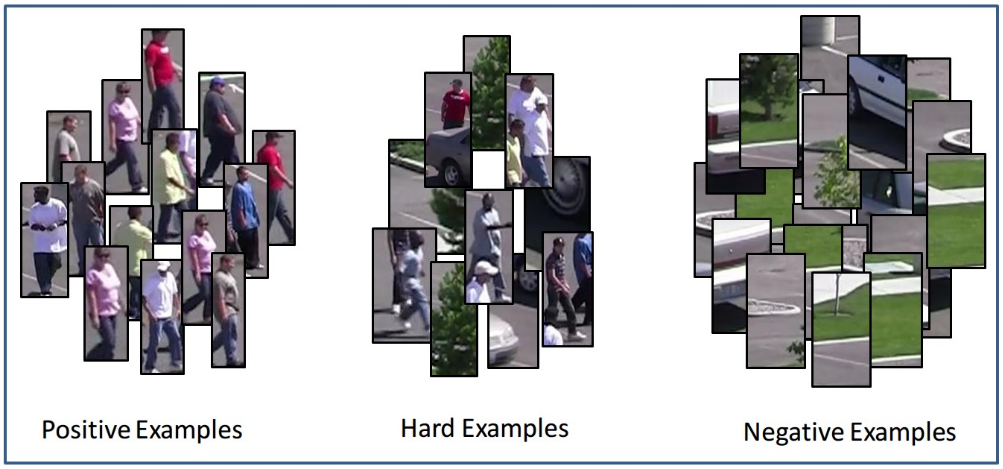


Figure 3.4: The three clusters show the detection Splitting results.

3.2.2 Classification

In the previous step, we discovered positives $\mathbf{P}$ and negatives $\mathbf{N}$ which we denote as an initial training dataset $\mathbf{T} = (\langle x_1, y_1 \rangle, ..., \langle x_n, y_n \rangle)$, where $y_i \in \{-1, 1\}$ are the binary class labels. Next we learn a new classifier with this dataset, which can be formulated as learning weights $\boldsymbol{w}$ of a linear classifier:

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \left(\frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{n} L(\boldsymbol{x_i}, y_i; \boldsymbol{w}) \right), \tag{3.4}$$

where $r(\cdot)$ is a regularization term, $L(\cdot)$ is a loss function which penalizes misclassification and margin errors, and C is a constant which sets the relative importance of the two. In practice, we use $L2$ normalization for $r(\cdot)$ which corresponds to a linear SVM. We train the classifier with a new superpixel-based feature representation, which will be described in section 3.3.

Due to the uncertainty of the DPM detector, the initial labels of detection examples y_i are not always correct. Ideally, we should use only the most confident examples to train the classier, otherwise the classifier will be compromised by the errors. Therefore, we employ a conservative learning framework which iteratively updates the classifier and thus minimizes bad examples. The learning framework is formulated in Equation 3.5:

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \left(\frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{n} a_i L(x_i, y_i; \boldsymbol{w}) \right), \tag{3.5}$$

where the binary variables $a_i \in \{0, 1\}$ control which examples are used in training. In particular, we choose more confident positives and negatives using the thresholds t_{p2} and t_{n2}, respectively. We train the classifier and use it to predict the confidences for all detections including the hard examples. Again, new confident examples are selected using thresholds t_{p2} and t_{n2} and added to the training dataset. In this way, the decision boundary is gradually refined for each iteration.

We repeat the above process until all the example labels are unchanged. In our experiments, it usually takes 5 to 7 iterations. After all the hard examples are labeled, we can project the positive examples back into the image sequence and generate the detection output. Figure 3.5 illustrates the iterative process of the unsupervised learning. Given the positive and negative examples, we can train a SVM classifier and use it to classify the hard examples. The most confident hard examples are then added to the training data.

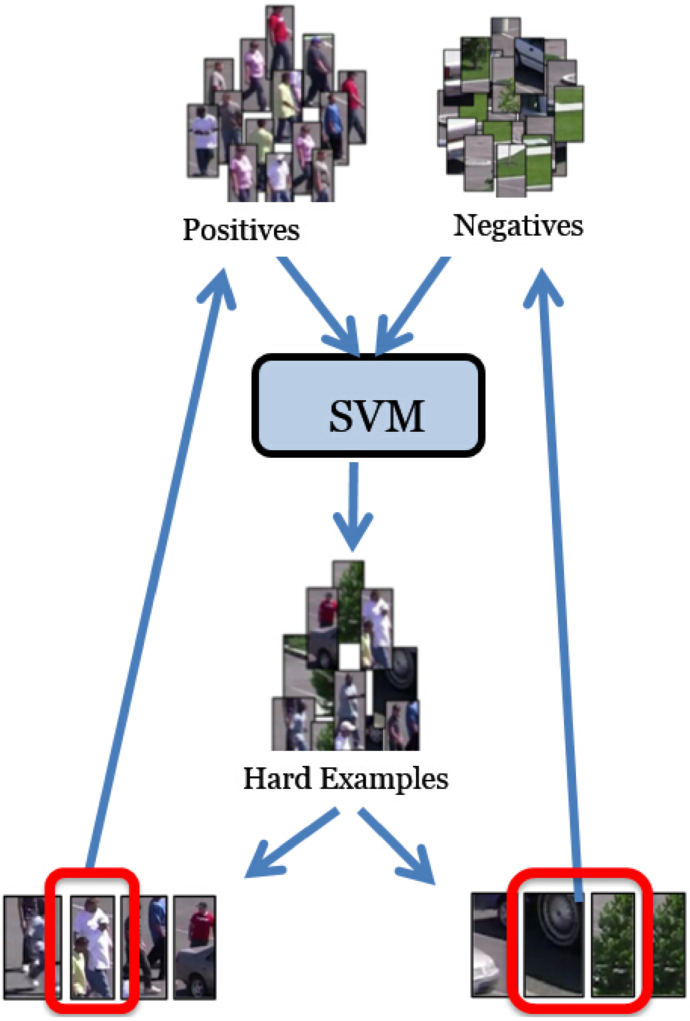


Figure 3.5: An illustration of unsupervised learning framework.

3.3 Superpixels-based Model

In this section, we introduce a new superpixel-based representation for human detection in videos. Most object detectors use HOG or Haar-like features which can represent a generic object class regardless of color variations. These features are very sensitive to pose variations. To handle

Algorithm 1 The proposed unsupervised learning algorithm

```
Input:
    Initial detections $\mathbf{D} = \{d_i\}$
    Confidence scores $c(d_i)$;
    Thresholds $t_p, t_n, t_{p2}, t_{n2}$
Output:
    Updated confidence scores, $c'(d_i)$;
Obtain the initial training set $\mathbf{T}$ using $\mathbf{D}, c(d_i), t_p, t_n$;
while $\triangle\mathbf{T} \neq 0$ do
    Train a SVM classifier $C_{svm}$ using $\mathbf{T}$;
    Obtain confidence scores $c'(d_i)$ using $C_{svm}$;
    Update the training set $\mathbf{T}$ using $\mathbf{D}, c'(d_i), t_{p2}, t_{n2}$;
end while
return $c'(d_i)$;
```

this problem, we propose a statistics appearance model with superpixels as units.

Superpixels have been successfully applied in many applications such as image segmentation [4], object localization [47] and tracking [102]. As the middle-level feature, it is a more natural representation than the pixel-grids. Using superpixels for image analysis has many advantages, summarized below:

- It is perceptually meaningful: Superpixels enable us to measure the feature statistics on a perceptually meaningful sub-region that has consistent color or texture.

- It is computationally efficient: it reduces the complexity of an image from hundreds of thousands of pixels to only a few hundred superpixels.

- The pairwise constraints between superpixels can represent long-range spatial relations compared with pixel-grids, and this property allows us to build graph-based models in the higher level.

- It has great flexibility for non-rigid objects, which avoids the mis-alignment of the HOG and Haar-like features on variant human poses.

There have been a number of methods [45, 62, 70, 84, 97] to obtain a superpixel map from pixel-grids. In this work we employ a state-of-the-art superpixel segmentation method [4] called simple linear iterative clustering (SLIC), which can efficiently segment superpixels using a K-means clustering approach.

We segment each detection window into M superpixels using the SLIC superpixel segmentation algorithm in [1]. We choose an appropriate number N so that each superpixel is roughly uniform in color and naturally preserves the boundaries of objects. In this way, we significantly reduce the image complexity while the appearance details are still well represented.

In order to encode both color and spatial information into superpixels, we describe each superpixel SP_i by a 5-dimensional feature vector $f_i = (L, a, b, x, y)$, in which (L, a, b) is the average $CIELAB$ color space value of all pixels and (x, y) is the average location of all pixels within the region. While the (L, a, b) feature represents the color cue in the video, the (x, y) feature implicitly captures important structure information in detection windows, where the human is usually in the center.

Superpixels cannot be directly used as the features since they do not spatially correspond across frames. Therefore we propose a superpixel-based BoW model to represent a detection window. More specifically, we first put all the extracted superpixels in the superpixel feature space. Then a M-word vocabulary is learned by clustering all the superpixels in this space using the K-means algorithm. Later the superpixels are aggregated into an M-bin L2-normalized histogram for each example. Finally, each example is represented in an M-bin BoW histogram. Figure 3.6 shows the steps for making the superpixel-based representation.

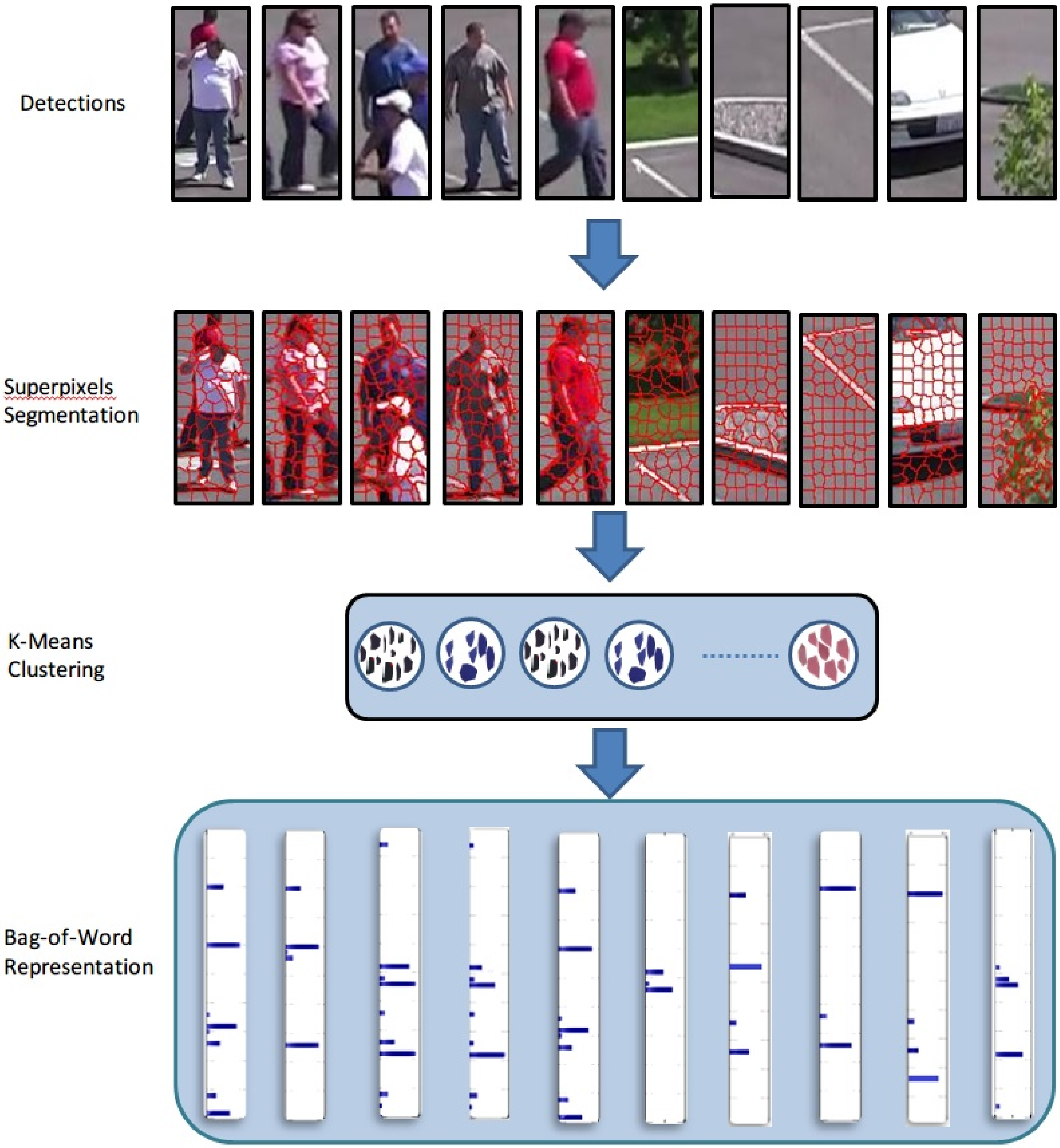


Figure 3.6: Superpixel-based representation. The detections are first segmented into superpixels. Next, all the superpixels are put together and clustered using the K-means method. Finally, each detection is represented by a histogram using the BoW method.

3.4 Experimental Results

We experimented extensively with the proposed method using four datasets: Pets2009 (PT), Oxford Town Center (TC) [14], PNNL-Parking Lot (PL) [92] and Skateboarding (SB). The experimental datasets provide a wide range of significant challenges including appearance change, occlusion, camera motion and cluttered background. In all the sequences, we only use the visual information and do not use any scene knowledge such as camera calibration or knowledge about static obstacles.

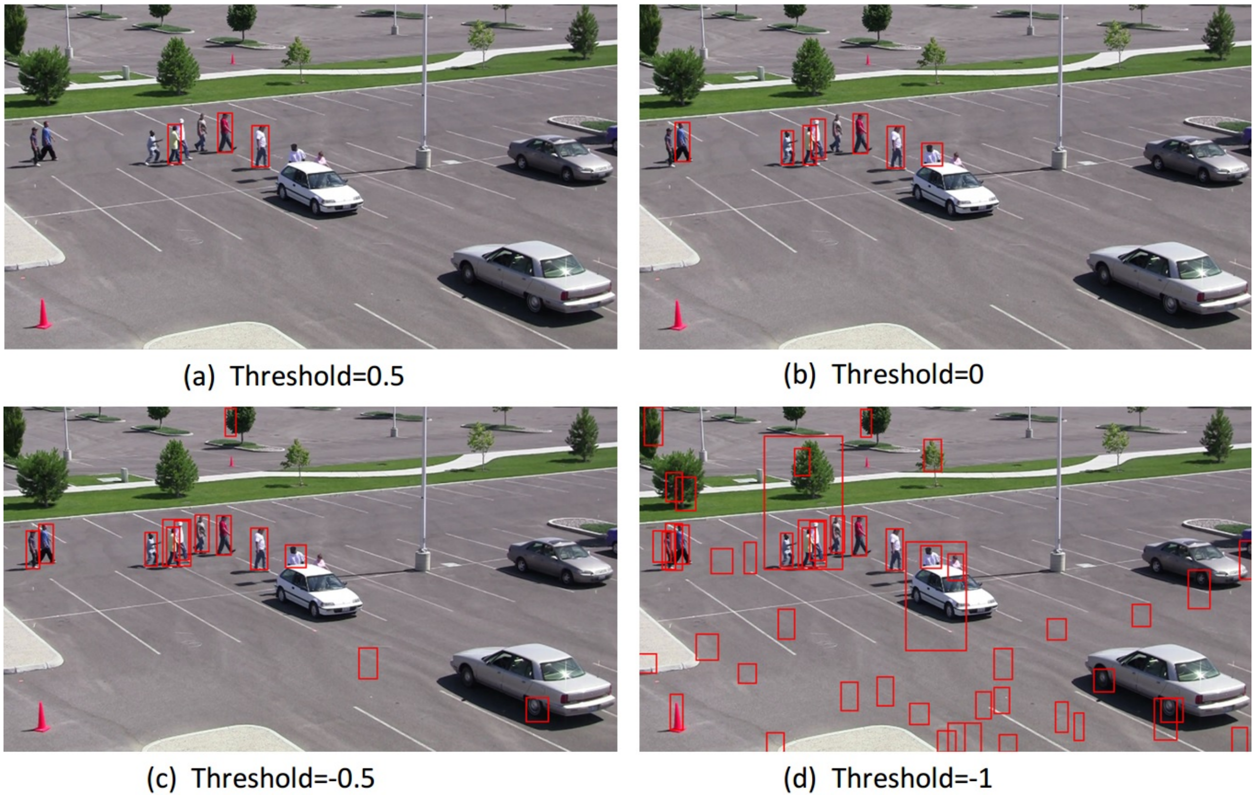
3.4.1 Implementation Details

In our implementation, the DPM human detector is pre-trained with the INRIA person dataset [33]. In the superpixels segmentation, all the detection examples are resized to 128×64. We set the number of superpixels for each example $N = 100$. In the K-mean clustering we set the size of codebook $M = 5000$.

It is important to choose appropriate thresholds for splitting the detections. Figure 3.7 shows results with different thresholds. With lower thresholds, the detector detect almost all true positives and significantly more false negatives. Based on the experiments we set the detection thresholds $t_p = 0$ and $t_n = -1$ in most cases. We set the classifier training thresholds $t_{p2} = 0.3$ and $t_{n2} = -0.7$.

3.4.2 Datasets

PNNL Parking lot Dataset (PL) We put together a high-resolution dataset which contains two video sequences of a parking lot using a static camera. These video sequences have 1,000 and 1,533 frames respectively, each of size 1920×1080. We manually annotated the ground-truth for these video sequences. The first video (PL1) is a moderately crowded scene including groups of pedestrians walking in queues with parallel motion and similar appearance. The second video



(a) Threshold=0.5

(b) Threshold=0

(c) Threshold=-0.5

(d) Threshold=-1

Figure 3.7: DPM detection results with different thresholds. The red bounding boxes represent the detections with a confidence score above the threshold. With lower thresholds, the detector detect almost all true positives and significantly more false negatives.

(PL2) is a more challenging sequence due to the large number of pose variations and occlusions, hence the results on this dataset are lower than for other datasets. However, our approach still performs significantly better than the DPM detector and HOG feature.

Town Center Dataset (TC) This is also a high-resolution video sequence consisting of 7501 frames, each of size 1920×1080. It was released with annotated ground-truth, we select the first 300 frames as our test video clip. This is a semi-crowded sequence with rare long-term occlusions. The motion of pedestrians is often linear and predictable. However, it is still quite challenging because of the inter-object occlusions and the difficulty of detecting pedestrians on a bike or with a stroller.

PETS2009 Dataset (PT) This dataset has a resolution of 768×576. We select one video sequence $(ID : S2_L1_Time12 - 34_View001)$ which has 795 frames as a test video. This is a relatively sparse scene including a few people walking in random directions. We up-sampled the resolution of this video by 2 because the original low resolution is not suitable for the DPM detector.

Skateboarding Dataset (SB) This dataset consists of two video clips captured by a hand-held camera. The first video has 367 frames, each of size 1280×720, the second video has 769 frames, each of size 640×480. This is a very challenging dataset due to the camera motion and extremes changes in pose.

3.4.3 Baseline Comparisons

We compare the proposed approach with two baselines.

Orig This baseline is the original DPM detector trained with the INRIA person dataset.

Ours-HOG This baseline uses the proposed unsupervised learning framework with HOG features. In the HOG implementation, each detection window is represented by a standard 3780-dimensional feature vector as in [33]; the other steps are identical to our proposed approach.

Ours-SP This is the proposed approach with superpixel-based features.

3.4.4 Detection Performance

We evaluated the detection performance on 6 video sequences from the four datasets. First, we analyzed the detector performance by computing Precision-Recall curves for all test videos, as shown in Figure 3.8. We used the criterion of the PASCAL VOC challenge [39] for evaluations in our experiments. As such, a detection that has more than 0.5 overlap with the groundtruth is determined as true positive.

In Figure 3.8, the proposed unsupervised approaches (Green and red dotted lines) outperform the original DPM detector (blue dotted line) by a significant margin. The precision of the three approaches is comparable at a low recall (high threshold). However, the precision of the DPM detector drops significantly when the recall increases above 0.9, while the unsupervised approaches maintains relatively high precision. The proposed superpixel-based representation is generally better than the HOG in all the testing videos. Since the initial DPM detector has already used HOG features, the HOG baseline doesn't obtain much new information from the examples. In contrast, the proposed superpixel-based feature unearth additional color cues, which is an important complement to the HOG features.

Based on the ROC curves, we used two types of measurement to compare performance of different methods. The first one is average precision (AP) [39], which summarizes the characteristics of the Precision-Recall curve in one number and is defined as the mean precision at a set of equally spaced recall levels. We chose the levels to be the same as [39] to represent the overall performance on each video, as shown in Table 3.1. Another measurement is the break-even point (BEP), which is the point at which precision equals recall. Table 3.2 shows the break-even points for all the testing datasets.

We observed that the proposed method outperforms the DPM detector as well as the unsupervised method with HOG in both tables. Table 3.1 shows that the proposed approach with superpixels achieves the best AP of all three. It improves the DPM detector by 5% to 15%, with an average of 10.35% over all test videos. The unsupervised method with HOG also improves the DPM detector by 8.95% on average, which demonstrates the generalization of the proposed unsupervised framework.

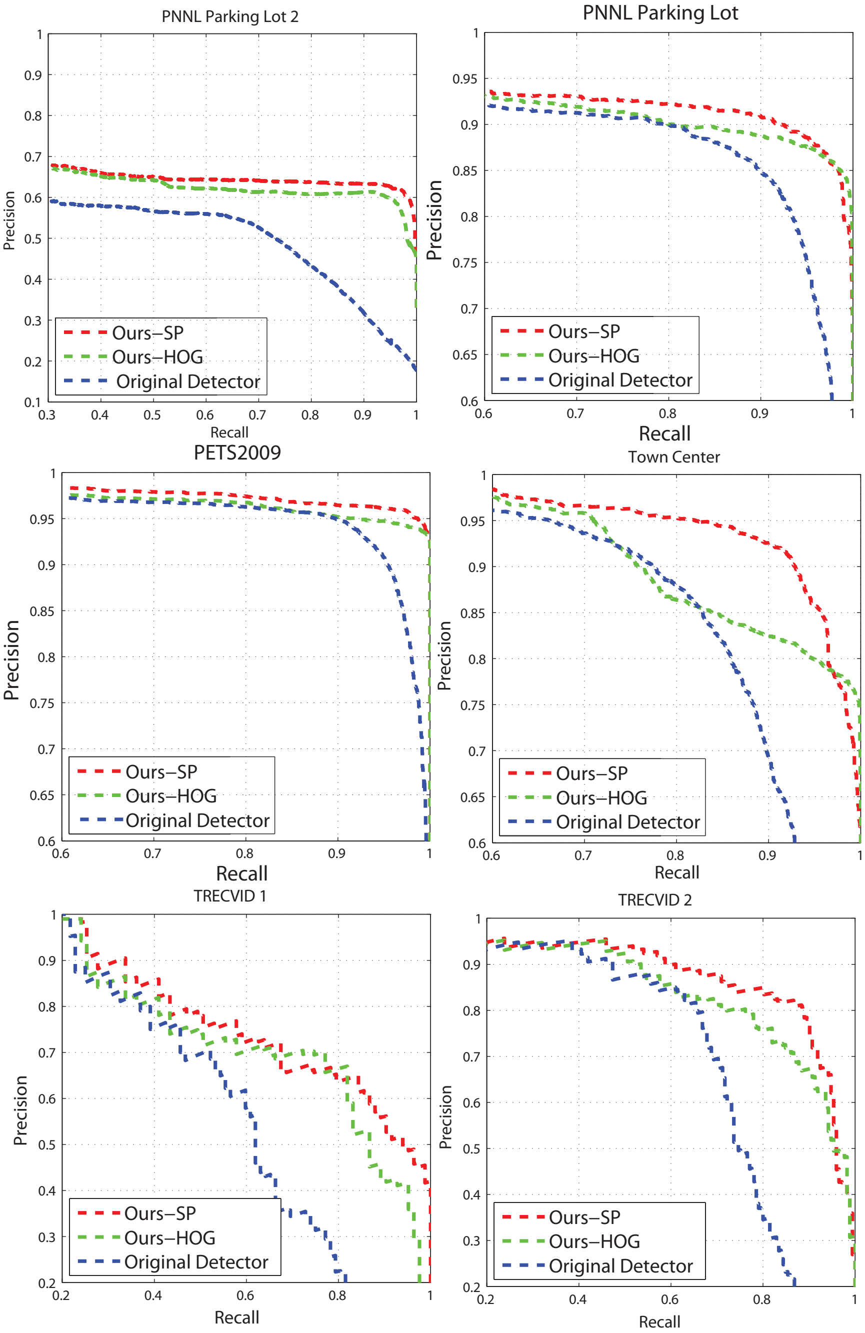


Figure 3.8: We compared our method against the original DPM detector once by choosing HOG features and once using the superpixel-based model. The blue line is the DPM detector, the green line is our unsupervised learning method using HOG features, and the red line is the proposed method.

Table 3.1: Average precision on our testing datasets. The second row shows the results of our method using HOG as descriptors and the third row shows the proposed method using the superpixel-based model.

Dataset	PL1	PL2	TC	PT	SB1	SB2
DPM	86.4	55.1	86.9	93.7	59.9	70.6
Ours w/o SP	91.6	66.1	93.6	97.9	73.8	83.3
Ours	**93.0**	**67.6**	**94.7**	**98.0**	**75.8**	**85.6**

Table 3.2: The precision-recall breakeven point.

Dataset	PL1	PL2	TC	PT	SB1	SB2
DPM	87.0	56.0	83.6	92.9	60.4	69.6
Ours w/o SP	88.7	62.0	84.7	94.5	67.9	78.3
Ours	**90.6**	**64.2**	**91.7**	**96.1**	**69.4**	**82.9**

Table 3.3: Average computing time (in seconds) for each frame. The first row is the DPM detector and the second row is our approach including the DPM detection step. Tested using a 3GHz CPU.

Dataset	PL1	PL2	TC	PT	SB1	SB2
DPM	7.1	6.9	7.7	1.5	3.4	1.3
Ours(without DPM)	1.4	1.5	1.6	0.4	0.7	0.3

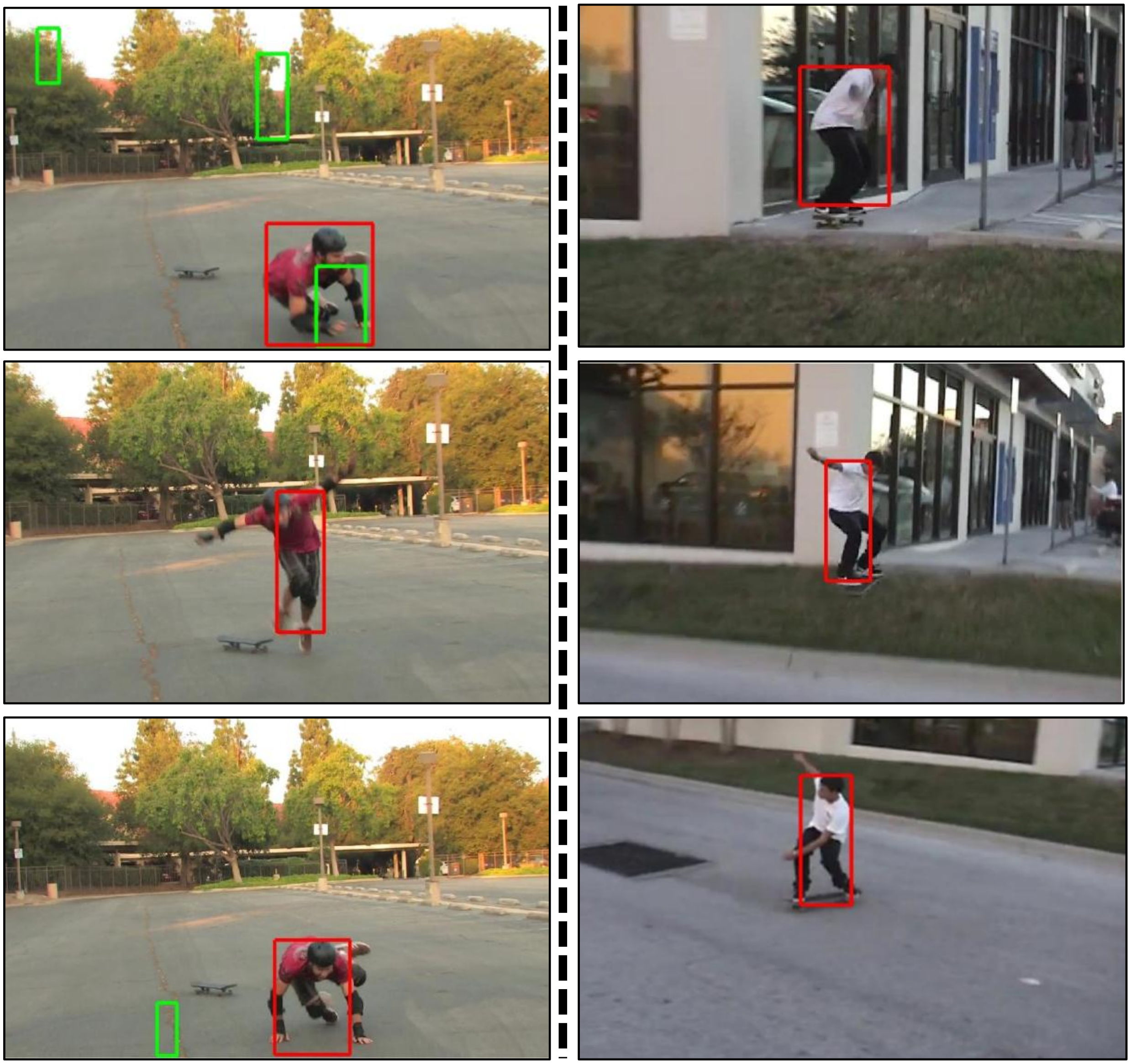


Figure 3.9: Detection results of the Skateboarding (SB) dataset. Each column shows three frames from one video clip. The green bounding boxes are the output by the DPM detector; the red bounding boxes are the output by our approach. It is clear that our approach has fewer false positives as well as false negatives.

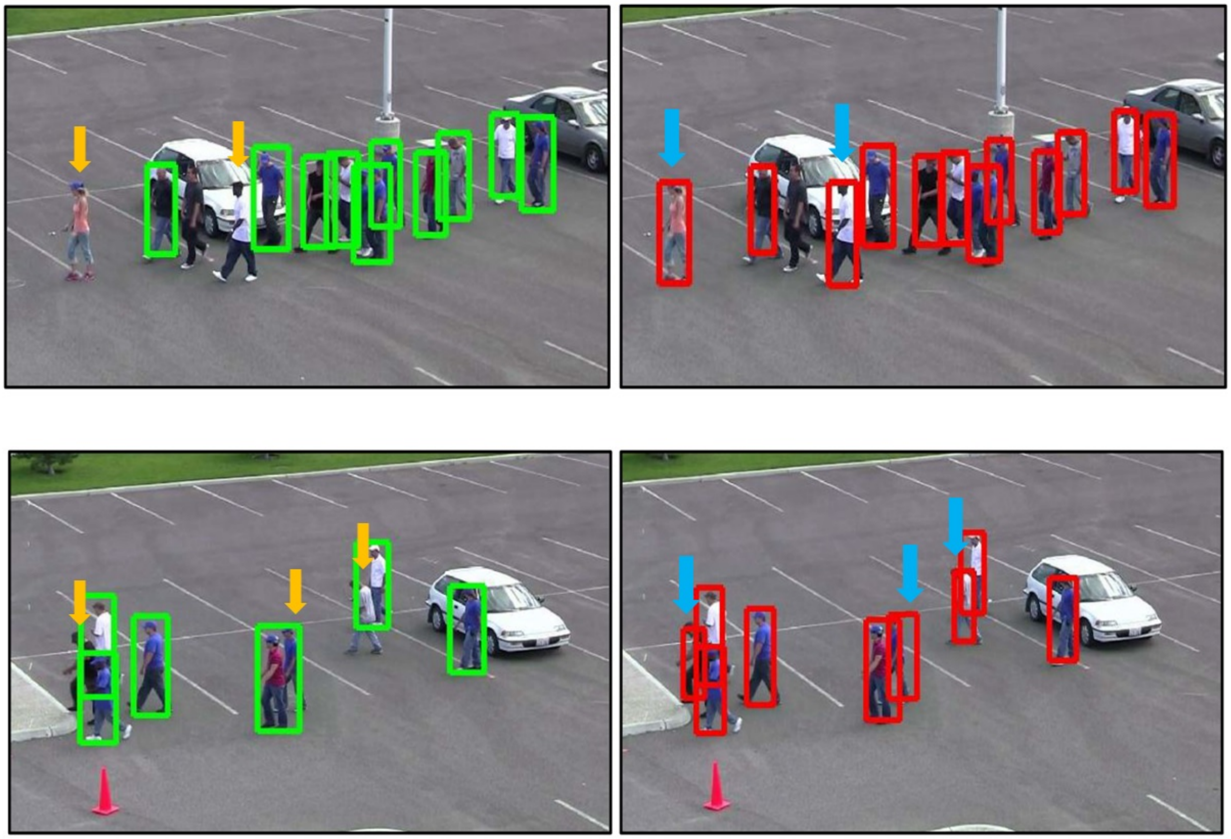


Figure 3.10: Detection results on PNNL Parking Lot 1 datasets. Each row shows the same frame detected using the DPM and our approach respectively. The green bounding boxes are the output by the DPM detector, and the red bounding boxes are the output by our approach. The yellow arrows show the mistakes by DPM detector, and the blue arrows show the improvements by our approach. It is clear that our approach gets better detection results.

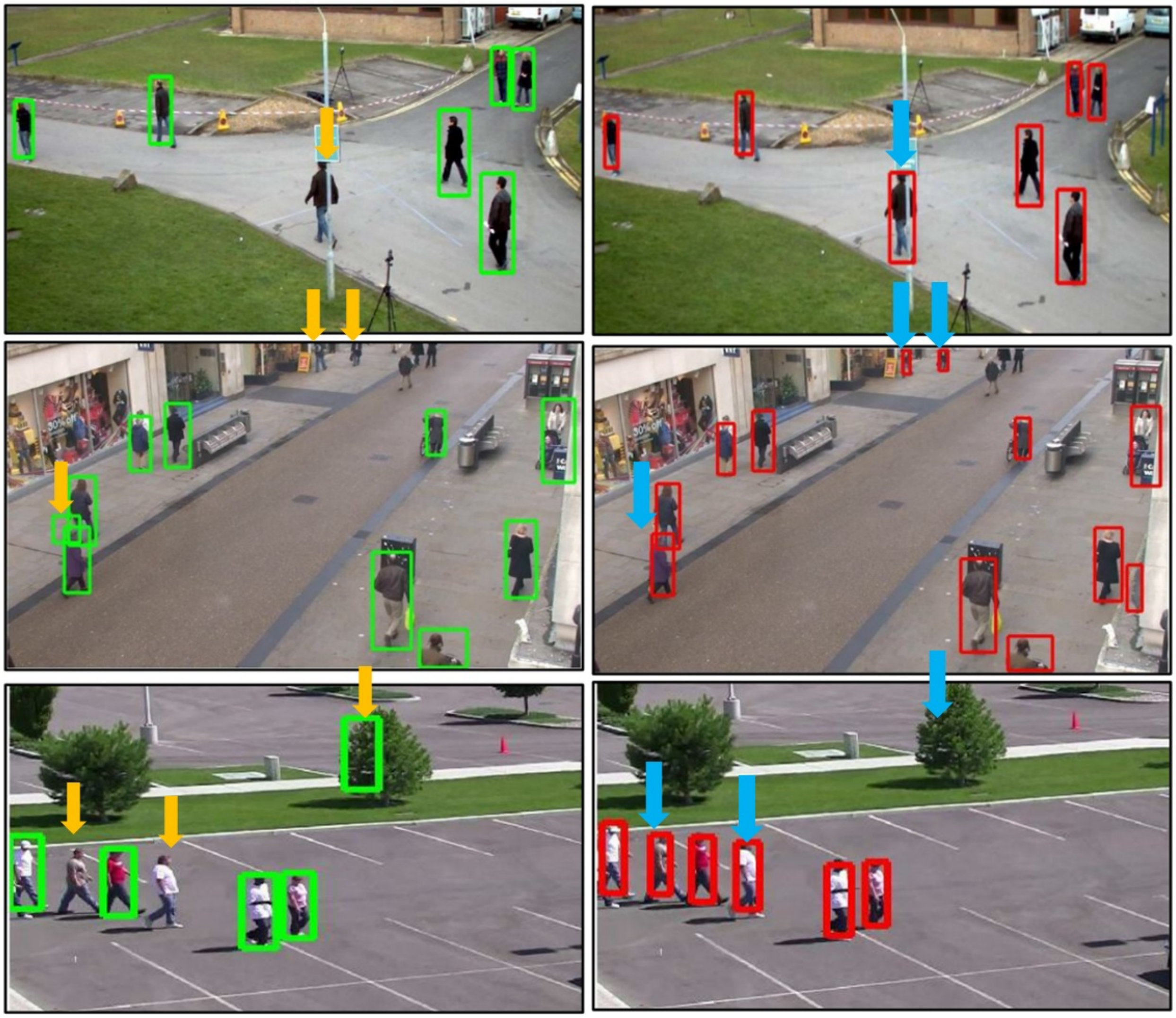


Figure 3.11: Detection results on several other datasets. The datasets from the first row to the last row are: Pets2009, Town Center and PNNL Parking Lot 2. The left column shows the results by the DPM detector, and the right column shows the results by our approach. It is clear that our approach has fewer false positives as well as false negatives.

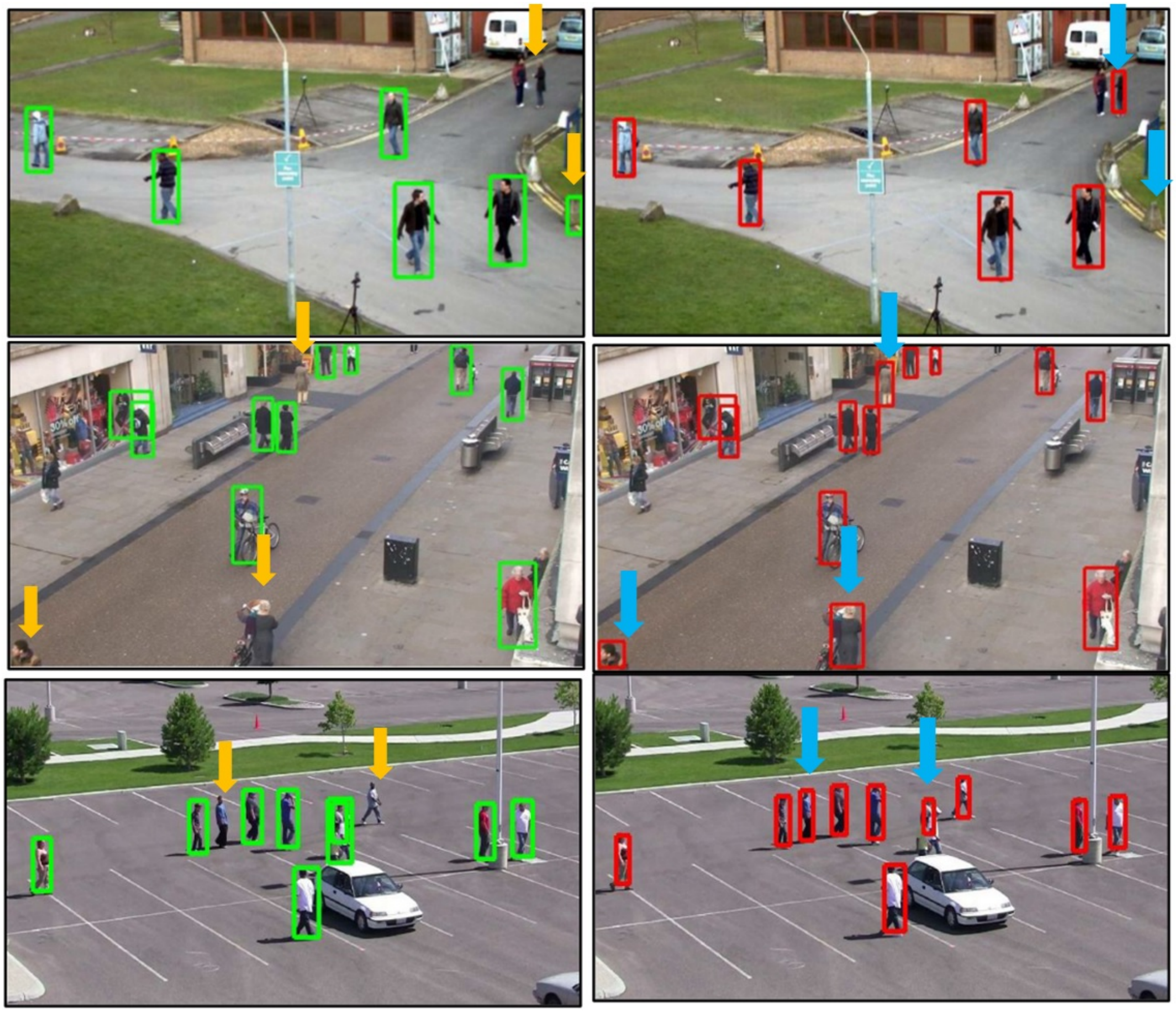


Figure 3.12: More detection results on several other datasets. The datasets from the first row to the last row are: Pets2009, Town Center and PNNL Parking Lot 2. The left column shows the results by the DPM detector, and the right column shows the results by our approach. It is clear that our approach get better detection results.

In the PT dataset, the original DPM detector has already achieved satisfying results but our approach performs even better. We observe that the HOG and Superpixels features perform almost the same in the low-resolution PT dataset, which demonstrates the superpixels are more sensitive to image resolution than HOG. In the SB dataset, our approach significantly improves the AP by 15%. The reason is that our approach can handle the extreme pose changes in this dataset. Figure 3.9 shows the qualitative detection results on the SB dataset. 3.12 shows the qualitative detection results on the other 3 datasets.

In addition, the computational cost of our approach is relatively low. Table 3.3 compares the average computing time for each frame between the DPM detector and our method. We observe that our approach takes 20% more time than DPM detector on average, which means the proposed unsupervised learning method adds only 20% running time to the DPM detector. The computational efficiency renders our method a fast and easy optimization for any application that requires detection.

3.5 Summary

In this chapter we proposed an effective method to improve a generic human detector using a superpixels-based BoW model. Our method captures rich information about individuals by superpixels; hence it is highly discriminative and robust against appearance changes. We employ a part-based human detector to obtain initial labels and gradually refine the detections in an iterative way. We demonstrated by experiments that our method effectively improves the performance of object detectors in four recent datasets.

CHAPTER 4: PART-BASED MULTIPLE-HUMAN TRACKING WITH OCCLUSION HANDLING

In the previous chapter we proposed an algorithm to improve the performance of a DPM human detector. The output of our detection approach can serve as a reliable initialization for the task of human tracking. Since the flexible part-based model can represent humans well, it is naturally a good choice for tracking pedestrians.

In this Chapter, we propose a part-based multiple-human tracking framework with online-learning. Our approach learns part-based, person-specific SVM classifiers which capture articulations of moving human bodies with dynamically changing backgrounds. With the part-based model, our approach is able to handle partial occlusions in both the detection and the tracking stages. In the detection stage, we select the subset of parts which maximizes the probability of detection. This leads to a significant improvement in detection performance in cluttered scenes. In the tracking stage, we dynamically handle occlusions by distributing the score of the learned person classifier among its corresponding parts, which allows us to detect and predict partial occlusions and prevent the performance of the classifiers from being degraded. Extensive experiments using the proposed method on several challenging sequences demonstrate state-of-the-art performance in multi-person tracking.

The rest of the chapter is organized as follows: In Section 4.1 we introduce the motivation for and overview of our approach. In Section 4.2, we describe an extended human detection method for partial occlusion handling. In Section 4.3, we present our part-based tracking framework with online-learning. In Section 4.4, we introduce a dynamic occlusion handling method for tracking. Experimental results are presented in Section 4.5.

4.1 Introduction

Object tracking is one the most fundamental problem in computer vision, especially for video surveillance. A large number of methods have been proposed in the last few decades. Several challenges make this problem very difficult: Firstly, the appearance of the target is often constantly changing in the field of view of the camera. Secondly, targets are often occluded in the field of view. Therefore, a successful tracker needs to associate the observations. Thirdly, targets often become occluded by other targets in the scene. Therefore, traditional motion-based trackers [30, 91] suffer in such scenarios. On the other hand, tracking with discriminative features has been extensively studied recently [9, 11, 28, 35, 49, 54]. In such methods, a specific classifier is trained with discriminative features in an online-learning fashion and then used to locate the target in consecutive frames. However, the online learned detectors will drift in long-term tracking with occlusions.

The ultimate goal of human tracking is to automatically initialize and track all the humans in the scene. Several sophisticated methods [13, 21, 56, 111] were proposed to tackle multi-target tracking by optimizing detection assignments over a temporal window, given certain global constraints. Such methods employ offline-trained detectors to find the targets and associate them with the tracks. Although they can handle several difficulties such as the uncertainty in the number of targets, occasional occlusions, and template drift in the long term, they still suffer when faced with appearance changes and occlusion. In particular, when tracking a crowd of pedestrians, the data association often fails in the aforementioned approaches due to pose variations, partial occlusions and background changes.

In this work, we address such difficulties by proposing a part-based representation in a tracking-by-detection framework. While the deformable part-based model [44] has shown excellent performance in static images, its potential has not been fully explored in tracking problems. Moreover, the availability of inexpensive high-definition sensors for surveillance provides the opportunity to exploit such detailed appearance representations.

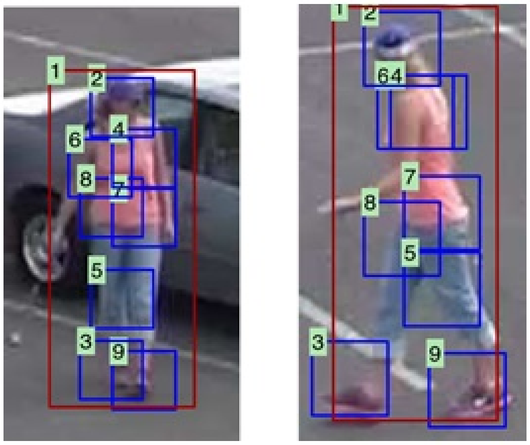


Figure 4.1: The part-based representation allows detailed correspondence between two images of the same person.. The figure demonstrates how the parts of two instances of a human correspond well even with significant appearance changes. Additionally, the background components in the detection boxes are automatically excluded from the correspondence.

Most tracking-by-detection methods use the whole detection window as an observation model. In contrast, we leverage the knowledge that all targets of one class (humans in this context) have a similar part structure; thus, we employ the sets of detected parts as observation models. Therefore, our method has several advantages: Firstly, the combination of parts provides a rich description of the articulated body; thus, it represents the human better than a single detection window. In particular, since the spatial relations of the parts in an articulated body are often flexible, corresponding targets using a holistic model (one detection box) is error-prone and may compare dissimilar parts of the body. In contrast, a part-based representation allows parts to be strictly compared to their corresponding parts. An example is shown in figure 4.1, where the parts of two instances of a person are well-corresponded even with different poses and backgrounds. Secondly, the part-based model excludes most of the background within the detection window and thus avoids confusion from background changes. Finally, since the part-based detector is offline trained by latent SVM using a large number of training samples, it captures a significant amount of discriminative information, which is essential for tracking.

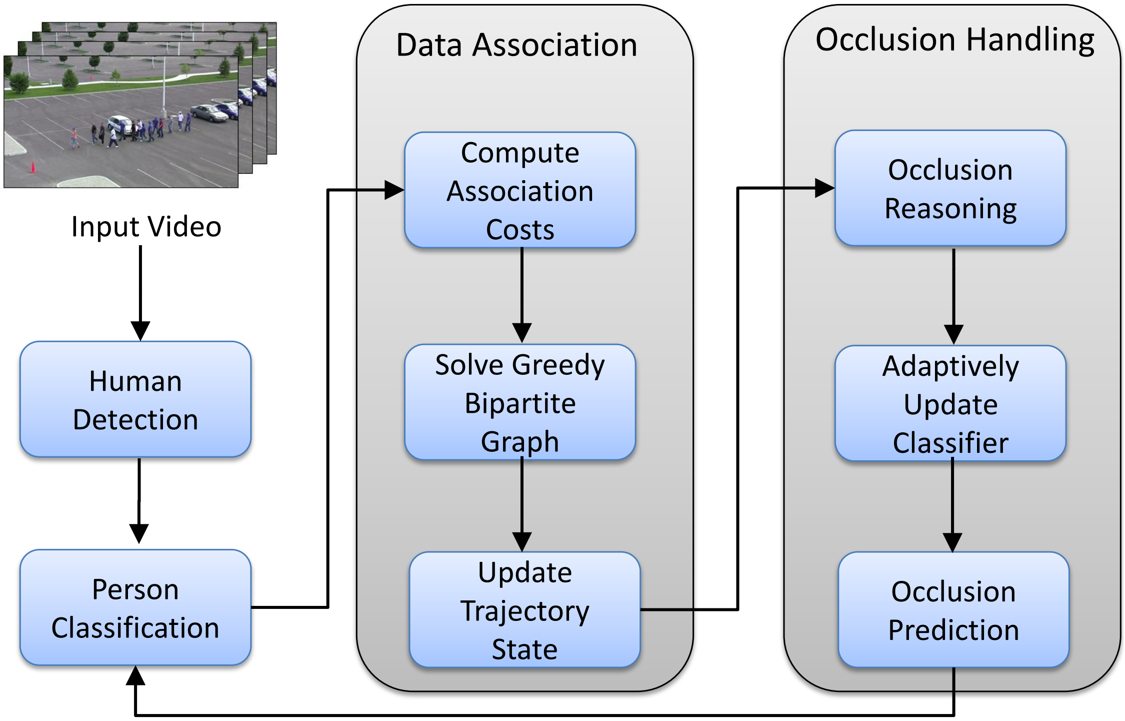


Figure 4.2: The flowchart of our tracking approach.

Our tracking algorithm consists of the steps illustrated in figure 4.2. First, we apply an extended part-based human detector to every frame of a video to detect all pedestrians, including partially occluded ones. Next, person-specific SVM classifiers are trained using the detections, and consequently used to classify the new detections. Next we use a greedy bipartite algorithm to associate the detections with the trajectories where the association is evaluated using three affinity terms: position, size, and the score of the person-specific classifier.

Additionally, during tracking, we consider the partial occlusion of a person using a dynamic occlusion model. In particular, partial occlusions are learned by examining the contribution of each individual part through a linear SVM. This inferred occlusion information is used in two ways: First, the classifier is adaptively updated with only the non-occluded parts, which prevents it from being degraded during the occlusion period. Second, the discovered occlusion information is passed to the next frame in order to penalize the contribution of the occluded parts when applying the person classifier.

4.2 Human Detection with Partial Occlusion Handling

The part-based representation also allows us to handle partial occlusions in crowded scenes. Since the combination of parts is very flexible, a person with partial occlusion may still have a few parts visible. In the detection stage, we select the subset of parts that maximizes the probability of detection, which significantly improves the detection performance.

We employ a deformable part-based model for human detection similar to [44]. However, the detector suffers when the human is occluded. In particular, the final score in [44] is computed from all the parts, without considering that some parts can often be occluded. Let H be the HOG feature of the image, and $p = (x, y)$ denotes a part specified by its position. The detection score at location (x_o, y_o) is computed in [44] as

$$score(x_o, y_o) = b + \sum_{i=1}^{n} s(p_i), \tag{4.1}$$

where b is a bias term, n is the number of parts, and $s(p_i)$ is the score of part i which is computed as

$$s(p_i) = F_{p_i} \cdot \phi(H, p_i) - d_{p_i} \cdot \phi_d(d_x, d_y), \tag{4.2}$$

where F_{p_i} is the part filter, and $\phi(H, p_i)$ denotes the vector obtained by concatenating the feature vectors from H at the subwindow of the part p_i. (d_x, d_y) is the displacement of the part with respect to its anchor position, $\phi_d(d_x, d_y) = (d_x, d_y, d_x^2, d_y^2)$ represents the deformation features, and d_{p_i} specifies the coefficients of the deformation features. Under this formulation, it is clear that even if the part was occluded, its corresponding score still contributes in the final detection score. This is a significant drawback especially when dealing with crowded sequences as shown in figure 4.3. In the figure, some humans appear fully in the image; however, several humans appear as only upper parts, or even only heads. Such impediment in [44] led previous works such as [13] and [85] to rely on only head detection and ignore the rest of the body.

To address this problem, we propose to infer occlusion information from the scores of the parts and consequently utilize only the parts with high confidence scores. Instead of aggregating the scores from the set of all the parts $P = \{p_0 \dots p_n\}$, we select the subset of parts $S = \{p_k \dots p_l\} \subseteq P$, which maximizes the detection score

$$score(x_o, y_o) = b + \arg\max_{S_m} \frac{1}{|S_m|} \cdot \sum_{i \in S_m} \frac{1}{1 + \exp(A(p_i) \cdot s(p_i) + B(p_i))},$$

where $|S_m|$ is the set cardinality, and the sigmoid function is introduced to normalize the scores of the parts. The parameters A and B are learned by the sigmoid fitting approach in [82]. Note that equation 4.3 corresponds to the average score of the parts in the subset. Since the average is sensitive to outliers, it is useful in capturing miss-detected parts. In other words, a subset S_m which contains occluded parts is likely to have lower average score than a subset without occluded parts. Therefore, by maximizing equation 4.3 we obtain the most reliable set of parts and its corresponding probability of detection, which we use as the final detection score. We consider only three possible subsets of parts, namely, head only, upper body parts, and all body parts. We found such subsets representative enough for most real world scenarios. Therefore, we do not need to search over all possible 2^n part combinations; instead, solving equation 4.3 involves only three evaluations which is a negligible overhead compared to the standard approach. Figure 4.3 demonstrates the advantage of our human detector over [44] in detecting occluded humans.

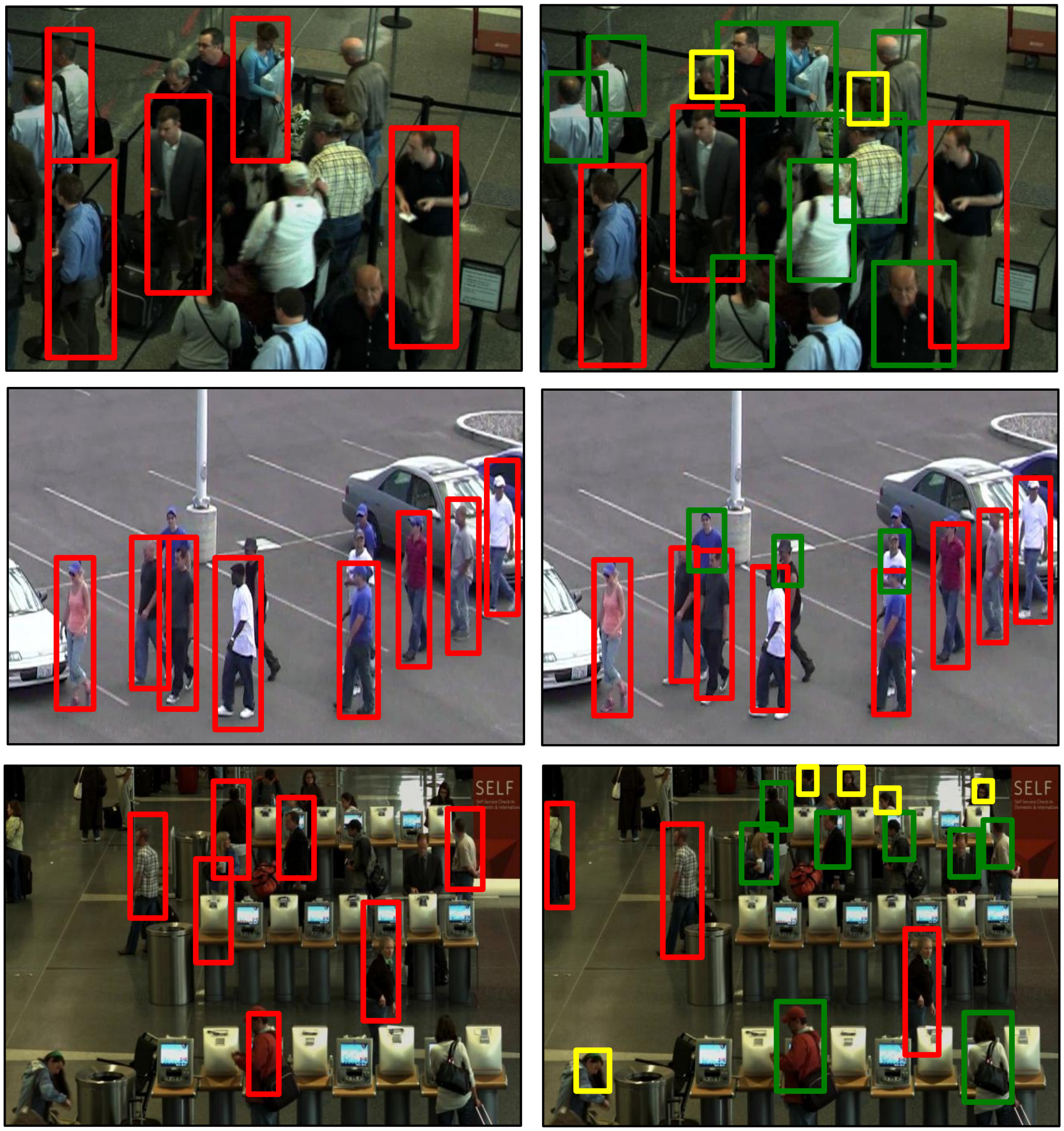


Figure 4.3: Results of our extended human detection method. Left: Human detection results using [44]. Right: Human detection results using our approach where red boxes show the humans detected as full bodies, green boxes show the humans detected as upper bodies, and yellow boxes show the humans detected as heads only. It is clear that [44] failed to detect occluded humans since it does not have an explicit occlusion model, while our approach detects the occluded parts and excludes them from the total detection scores, thus achieving significant improvements especially in crowded scenes.

4.3 Multiple Human Tracking with online-learning

In this section we introduce the proposed tracking algorithm with online-learning. In the tracking-by-detection framework, an essential step is to associate the detections across the frames by their similarities in terms of appearance, position and size. In a semi-crowded scene, it is often difficult to maintain the correct identities for multiple humans that have similar appearance. To address this problem, we employ online learned person-specific SVM classifiers to generate the appearance similarity scores.

4.3.1 Person-specific Classifier

We train an online person-specific classifier for each individual target. In each frame, the human detections are classified by the person classifiers. Relevant previous work mostly used an Adaboost classifier with Haar-like features; in contrast, our approach leverages the detected human parts and trains an SVM classifier. We extract features from each individual part, and then concatenate them in a fixed order as a feature vector. Figure 4.4 illustrates how to extract a feature vector from a part-based model.

We choose $CIELab$ color histogram and Local Binary Pattern [5] as features because they are highly discriminative for individuals and are complementary to the HOG feature which is used in the human detector. The classifier is trained using the detections included in each track of a person. In particular, the positive examples are taken from all detections in the trajectory and the negative examples are taken from the detections of the other trajectories augmented with random patches collected from the surrounding background in order to improve the classifier's discrimination in relation to the background.

Figure 4.5 shows two examples of tracking human with part-based model. We observe that the parts are well corresponded across the frames, therefore the part-based model is a suitable representation for tracking humans.

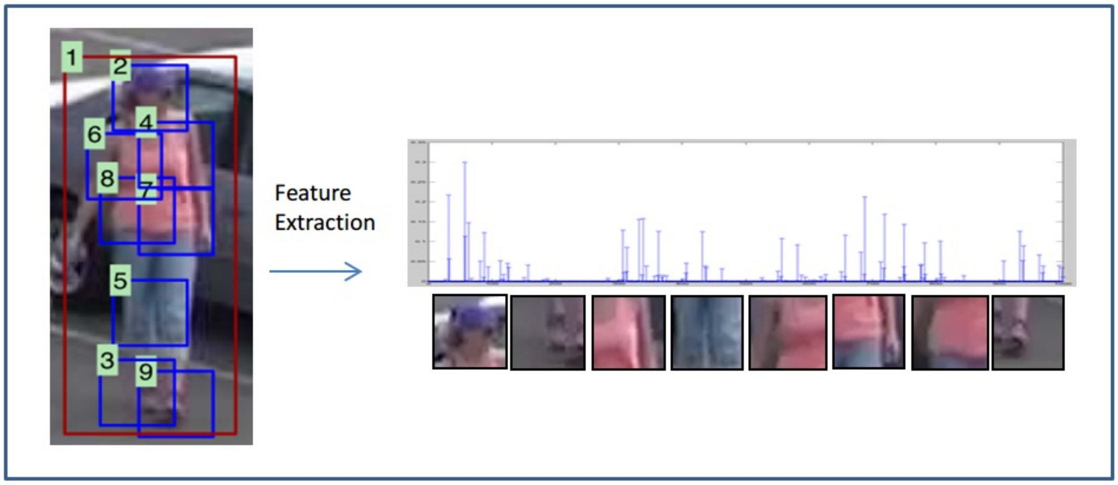


Figure 4.4: Feature extraction for a part-based model. The left image shows a DPM detection with parts. The right figure shows a concatenated feature vector and the corresponding parts.

4.3.2 Tracker Initialization

We represent a track of a person by a set of variables describing its state; namely, the position, the velocity, the size, the template image, the parts' positions, and the appearance feature vectors. The initialization is fully automated by the human detector. The detections which are not associated with any existing trajectories are used to initialize a new potential trajectory. An online-learning classifier is also initialized with the trajectory. To avoid false detections we measure the confidence of a potential trajectory m by

$$c(s) = \sum_{k=1}^{l} HI(m_k, m_{k-1}) c_d(m_k), \tag{4.3}$$

where $HI(m_k, m_{k-1})$ is the histogram intersection between two adjacent detections, l is the length of the trajectory, and $c_d(\cdot)$ is the confidence score from the class detector. Once the confidence of a continuous potential trajectory becomes larger than a threshold, it gets formally initialized, otherwise it gets deleted. We also terminate a potential trajectory that lasts longer than a threshold to avoid persisting false detections.

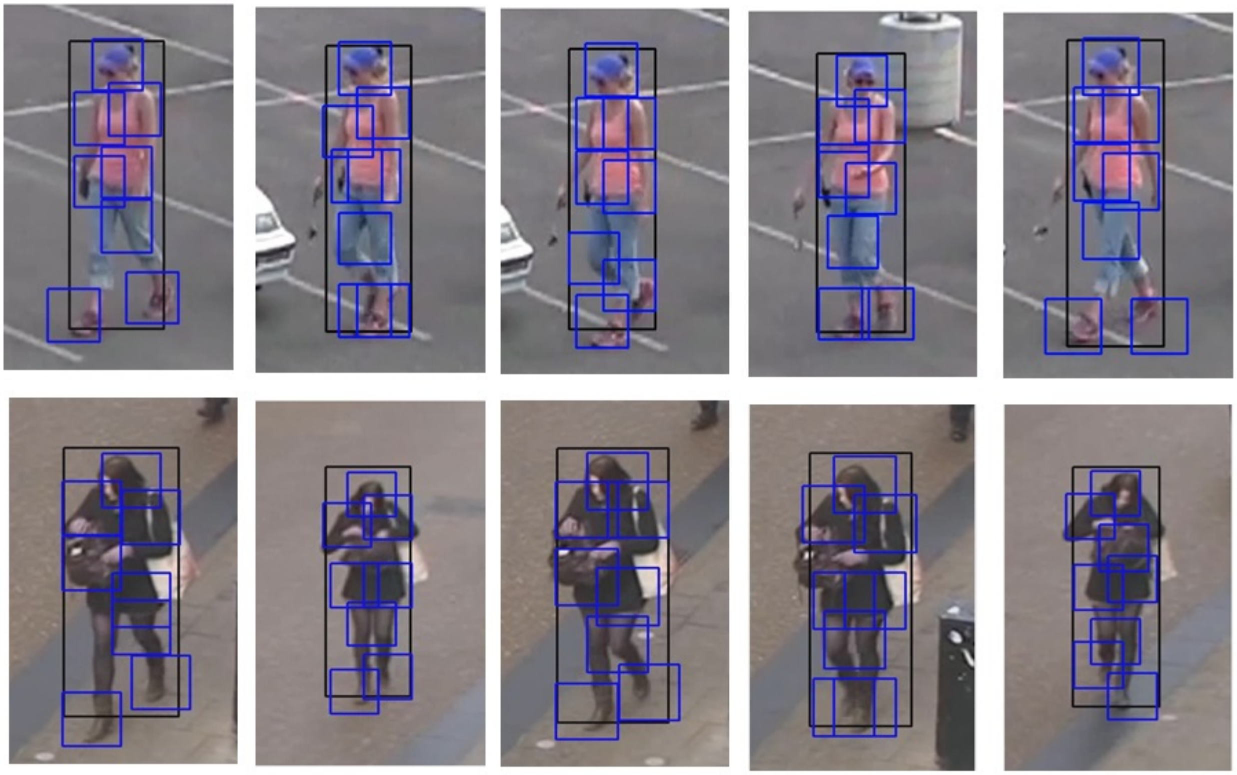


Figure 4.5: Two examples of tracking human with part-based model. Each row shows a pedestrian is being tracked through different frames of a video. In both examples the parts are well corresponded across the frames, which enables us to track humans robustly.

4.3.3 Data Association

Several tracking applications require online forward tracking, i.e. the current trajectories should depend only on previous frames, not on future observations. To meet such requirements, we use a first-order Markov model in data association, in which trajectories are continuously growing as the tracking proceeds. In every frame, the detections are associated with existing trajectories by a greedy bipartite assignment algorithm which has been used in [20, 105]. In particular, for each frame, we construct an affinity matrix M for the trajectories and the detections. Consequently, the pair with the maximum affinity $M_{i,j}$ is selected as a match, and the i-th row and the j-th column are deleted from M. This procedure is repeated until no more pairs are available. Finally we examine the affinity of all the matches and use only those with an affinity value larger than a threshold in order to ensure all matches are potentially correct. To evaluate the affinity of a trajectory i and a detection j, we use

$$M(i,j) = C_i(j) \cdot Z(i,j) \cdot E(i,j) \tag{4.4}$$

where the three terms Z, E, and C represent the affinities of position, size, and appearance respectively.

Position Affinity: We establish a linear motion model for each trajectory and use a Kalman filter to predict the potential position in the current frame. Consequently, the position affinity can be obtained by

$$Z(i,j) = \begin{cases} \exp(-\frac{d(i,j)^2}{\sigma_p^2}) & \text{if } d(i,j) < \tau(i) \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

where $d(i,j)$ is the spatial distance between the centroids of the detection i and the prediction for the next location of the trajectory j, and σ_p is a constant. $\tau(i)$ is a threshold which limits the search area, thus reducing the computational cost. A constant threshold $\tau(i)$ is not reliable since a trajectory may not get associated with any detections for a long time due to occlusion or mis-detection. Therefore, we apply an adaptive threshold

$$\tau(i) = \tau_o + v_i t_i, \tag{4.6}$$

where τ_o is the initial dimension of the search region, v_i is the current velocity of trajectory i

calculated by the linear motion model, and t_i is the time of being occluded from the last confident detection. The adaptive threshold will gradually enlarge the local search area when the trajectory is not matched to any detections.

Size Affinity: We measure the size affinity by

$$E(i,j) = \exp(-\frac{(g_i - g_j)^2 + (h_i - h_j)^2}{\sigma_s^2}), \tag{4.7}$$

where g and h are the width and the height of the detection bounding box, and σ_s is a constant. The size of a trajectory is represented by the size of the last detection in the trajectory.

Appearance Affinity: The appearance affinity is measured by the prediction of the online-learned classifier. We extract the features from the detection j and obtain $C_i(j)$ as the real-value output of classifier i.

4.3.4 Updating the State of the Trajectory

When a new detection is associated to a trajectory, we update all its state variables based on the new detection. However, when there is no detection associated due to occlusion or mis-detection, we use a correlation-based Kalman filter to track the head part of the target in the local area. In this case we only update the motion state variables, namely, the position and the velocity. This heuristic is particularly useful in crowded scenes where only humans' heads are observed. On the other hand, if a matching detection for the trajectory is later found, our tracker will immediately update its state according to the new detection. This allows us to handle long-term full occlusions with unpredictable motion patterns.

4.4 Dynamic Occlusion Handling in Tracking

When updating the state of a trajectory, a potential issue occurs if the person is partially occluded but still getting associated with the right trajectory, in this case the classifier will be updated with noise and its performance will gradually degrade. Therefore, we employ an occlusion reasoning method to handle this problem. Furthermore, with the occlusion information, we can predict occlusion in consecutive frames to improve the accuracy in data association. It was shown in [103] that in a detection window, occluded blocks respond to the linear SVM classifier with negative inner products. We adopt a similar approach to infer which parts are occluded using the online learned SVM.

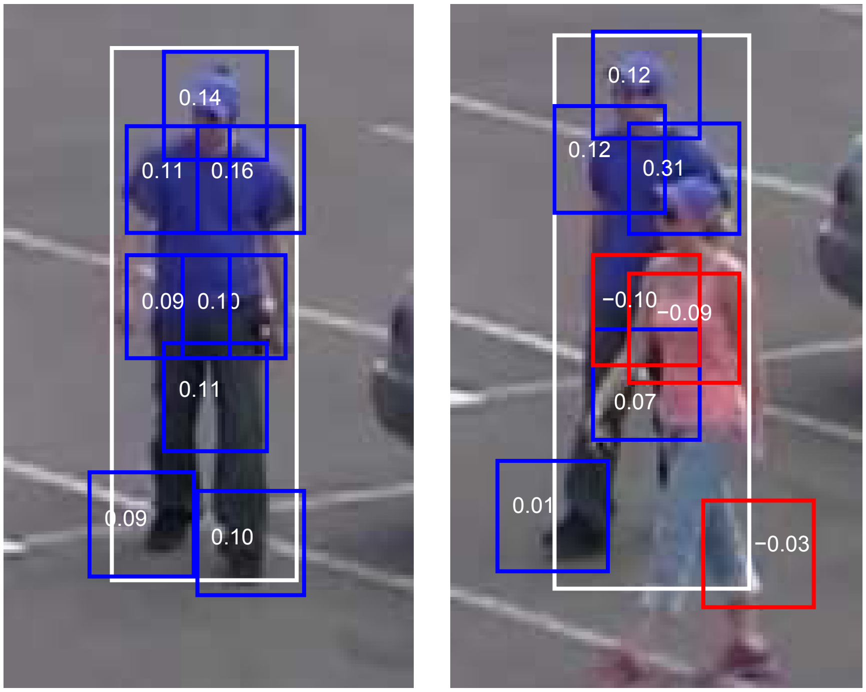


Figure 4.6: An example of our occlusion reasoning method. Left: a certain target human is detected, none of his parts are occluded (shown in blue); thus, all parts have positive responses. Right: the same target is observed again in another frame; however, three of his parts are occluded by another person (occluded parts are shown in red). The occluded parts have negative responses as shown in the figure. In our framework, the occluded parts are excluded when updating the person's model. Additionally, when evaluating the appearance affinity, the occluded parts have lower contribution since they are assigned weights relative to their score (refer to equation 4.11).

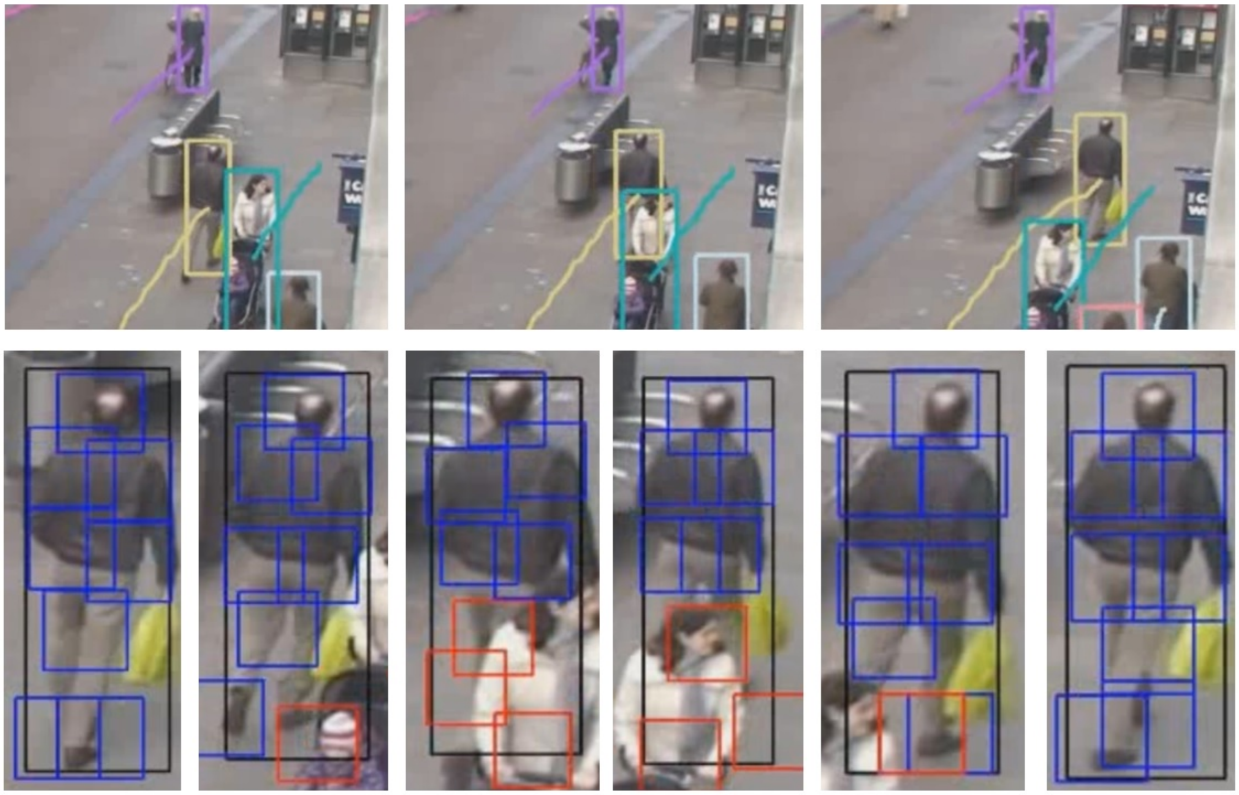


Figure 4.7: An example of our dynamic occlusion handling approach. Top row shows a man is partially occluded by a woman with a stroller, and bottom row shows the zoom-in images with visualized parts. The blue bounding boxes are non-occluded parts and red bounding boxes are occluded parts. It is clear that our method correctly detects the partial occlusion.

Assume that the testing feature vector $\mathbf{x}$ consists of n sub-vectors corresponding to n parts, written as $\mathbf{x} = (\mathbf{s_1}, \ldots, \mathbf{s_n})$. The decision function for linear SVM classifier is

$$f(x) = \beta + \sum_{k=1}^{l} \alpha_k \langle \mathbf{x}, \mathbf{x_k} \rangle = \beta + \mathbf{x}^T \mathbf{w}, \tag{4.8}$$

where $\mathbf{x_k}$ is a support vector and $\mathbf{w}$ is the weighted sum of all support vectors. We can divide $\mathbf{w}$ to n sub-vectors $\mathbf{w} = (\mathbf{w_1}, \ldots, \mathbf{w_n})$, and find a set of $\{\beta_i\}$ such that $\beta = \sum \beta_i$, then the separate contribution of each part is represented by

$$f(\mathbf{s_i}) = \beta_i + \mathbf{s_i}^T \mathbf{w_i}. \tag{4.9}$$

Each time we re-train a person-classifier, we calculate β_i in a similar way to [103] using the previously observed training samples. Consequently, we obtain the score for each individual part. The part with a negative score is mostly likely to be occluded or underwent significant appearance change. Therefore, when applying the data association for the trajectories, we examine the classifier scores and employ occlusion reasoning for all trajectories with low confidences. As such, the occlusion information is used to adaptively update the classifier by only extracting features from the parts with a high confidence score which are likely to correspond to the non-occluded parts, while the features for the occluded parts are obtained from the feature vectors of the previous frames. Using this technique, the occluded parts will not be included in updating the classifier. Figure 4.6 demonstrates how occluded parts have negative responses to the SVM. Figure 4.7 shows example results of our dynamic occlusion handling.

On the other hand, the occlusions are highly correlated in the adjacent frames of videos. Hence, when a partially occluded part is detected in one frame, it will have a high probability of being occluded in the consecutive frames. We harness such smoothness in occlusion to improve the classification performance by introducing an occlusion prediction method into the data association in order to improve the accuracy. First, we map the part SVM scores into occlusion confidence by

$$G(i) = \exp(\frac{f_t(\boldsymbol{x_i})}{\delta}), \tag{4.10}$$

where δ is a constant. The occluded parts will likely have lower scores as demonstrated in the example in figure 4.6. Therefore, when evaluating the appearance affinity during the data association, we weight each part by the occlusion confidence and normalize the total score

$$C = \frac{n \sum_{i=1}^{n} G(i) f_{t+1}(\boldsymbol{x_i})}{\sum_{i=1}^{n} G(i)}. \tag{4.11}$$

In the following frames, we examine the occlusion again and update the occlusion confidence until the classifier score is larger than a threshold. This allows the occlusion information to be passed across continuous frames, and the appearance affinity to have higher weight corresponding to non-occluded parts. Figure 4.8 shows example results of our occlusion reasoning in tracking.

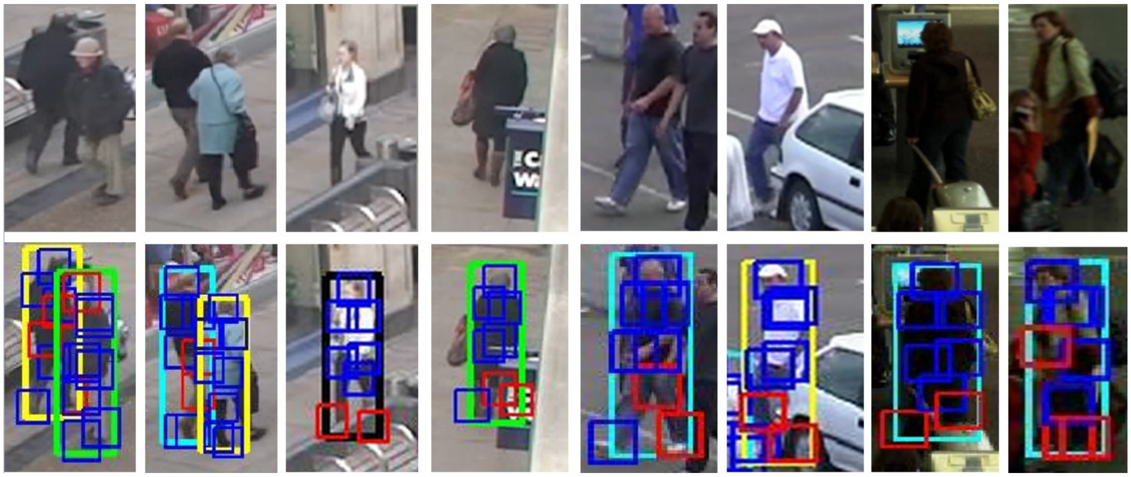


Figure 4.8: Examples of our dynamic occlusion handling approach. Top row shows the original image, and bottom row shows the detected humans and their corresponding parts, where the occluded parts shown in red.

4.5 Experimental Results

We experimented extensively on the proposed method using the Oxford Town Center dataset [13], and two new datasets that we collected; the Parking Lot dataset, and the Airport dataset. The experimental datasets provide a wide range of significant challenges including occlusion, crowded scenes, and cluttered background. In all the sequences, we only use the visual information and do not use any scene knowledge such as camera calibration or static objects. It is important to notice that we selected the aforementioned datasets since they include high quality imagery which is more suitable to our approach since the part-based model requires detailed body information.

In our implementation, we use the pre-trained pedestrian model with 8 parts from [44]. The feature vector for each part consists of a 125-bin RGB color histogram using 5 bins for each channel and a 59-bin LBP histogram. We apply normalization for each part and concatenate all 8 parts into one feature vector of $1,472$ dimensions. The training data for each person-specific classifier consists of up to 100 positive samples and 100 negative samples. When the number of collected samples exceeds this limit, we delete the oldest ones to ensure the model is up-to-date. Aside from the human detection, our tracker runs at 1 to 5 fps on a conventional desktop, depending on the number of humans in the sequence.

We evaluated our tracking results using the standard CLEAR MOT metrics. A CLEAR MOT metric is used to measure and compare the performance of multiple object trackers for a wide variety of tracking tasks. It consists of four scores: Detection Accuracy (DA) evaluates the performance with respect to false negatives and false positives. Tracking Accuracy (TA) evaluates the performance with respect to false negatives, false positives and ID-switches over all frames. Detection Precision (DP) is the overlap ratio for matched object-hypothesis pairs over all frames. Tracking Precision (TP) is similar to DP, but taking into account the splits and merges of tracks.

It is worth noting that TA has been widely accepted as the main gauge of performance of the tracking methods. TP and DP only evaluate the performance with respect to deviation errors. TA accounts for all object configuration errors made by the tracker, therefore it is the most important measure of the tracker's performance.

We show the quantitative results in Table 4.1, 4.2 and 4.3 for different dataset. We compare our method with two baseline settings and other methods in previous works. ”Ours1” is our tracking method without occlusion handling is the detection stage. ”Ours2” is our proposed method without occlusion handling in the tracking stage. We will analyze tracking performance by dataset. ”Ours3” is the proposed method.

Town Center Dataset: The frame resolution in this dataset is 1920×1080, and the frame rate of 25 fps. This is a semi-crowded sequence with rare long-term occlusions. The motion of pedestrians is often linear and predictable. In table 4.1, we compare results with [13, 59, 81, 107, 109] using the results reported in [59]. With the same experimental settings, our method significantly outperforms all previous methods in TA. The improvement in our method is a result of two main factors: First, the part-based model could better represent the articulated body and thus improves the accuracy in data association. Second, our dynamic occlusion handling module allows us to robustly track partially occluded humans. We observe that the proposed human detection method and the DPM detector are comparable in tracking performance. Since this dataset has extreme occlusions rare, the results of the two baseline methods (”Ours1” and ”Ours2”) are very close to the proposed method (”Ours3”).

Table 4.1: Tracking results on the Town Center Dataset. ”Ours1” is our tracking method without occlusion handling is the detection stage. ”Ours2” is our proposed method without occlusion handling in the tracking stage.

Dataset	MOTA	MOTP	MODA	MODP
Benfold et al. [13]	64.8	80.4	64.9	80.5
Zhang et al. [109]	65.7	71.5	66.1	71.5
Pellegrini et al. [81]	63.4	70.7	64.1	70.8
Yamaguchi et al. [107]	63.3	70.9	64.0	71.1
Leal-Taixe et al. [59]	67.3	71.5	67.6	71.6
Ours1-occ tracking	72.2	71.1	72.7	71.2
Ours2-occ detection	72.6	71.3	73.1	71.4
Ours3	72.9	71.3	73.5	71.4

Parking Lot Dataset: This dataset contains two video clips. The frame resolution in this dataset is 1920×1080, and the frame rate is 29 fps. This is a modestly crowded scene including groups of pedestrians walking in queues. The challenges in this dataset include long-term inter-objects occlusions, camera jittering, and similarity of appearance among the humans in the scene. The tracking results for the Parking Lot dataset are summarized in table 4.2. With extreme occlusions rare, the method with occlusion handling slightly outperforms other methods.

Table 4.2: Tracking results on the Parking Lot dataset.

Dataset	MOTA	MOTP	MODA	MODP
Ours1-occ tracking	77.1	73.7	77.5	73.8
Ours2-occ detection	78.9	74.1	79.3	74.2
Ours3	79.3	74.1	79.8	74.2

Table 4.3: Tracking results on the Airport Sequence.

Dataset	MOTA	MOTP	MODA	MODP
Ours1-occ tracking	27.2	66.1	28.4	66.3
Ours2-occ detection	49.7	67.2	50.1	67.4
Ours3	52.2	67.2	53.6	67.4

Airport Dataset: We use one sequence from the Airport Dataset that we collected from the surveillance videos at Boston Logan Airport. The frame resolution in this dataset is 4000×2672, and the frame rate is 5 fps. This is a very challenging real world scene with bad occlusions resulting from both static obstacles in the scene and inter-person occlusions. Additionally, the humans' appearance and pose significantly change during the sequence because of the wide field of view of the camera and the low frame rate. However, our approach still achieved promising results on this dataset. The tracking results for the airport dataset are shown in table 4.3. Note that TA is significantly higher using the proposed detection than using [44] on this dataset since it is more crowded, and thus occlusions occur very frequently. With occlusion handling in both detection and tracking stages, "Ours3" has a significant better performance in TA.

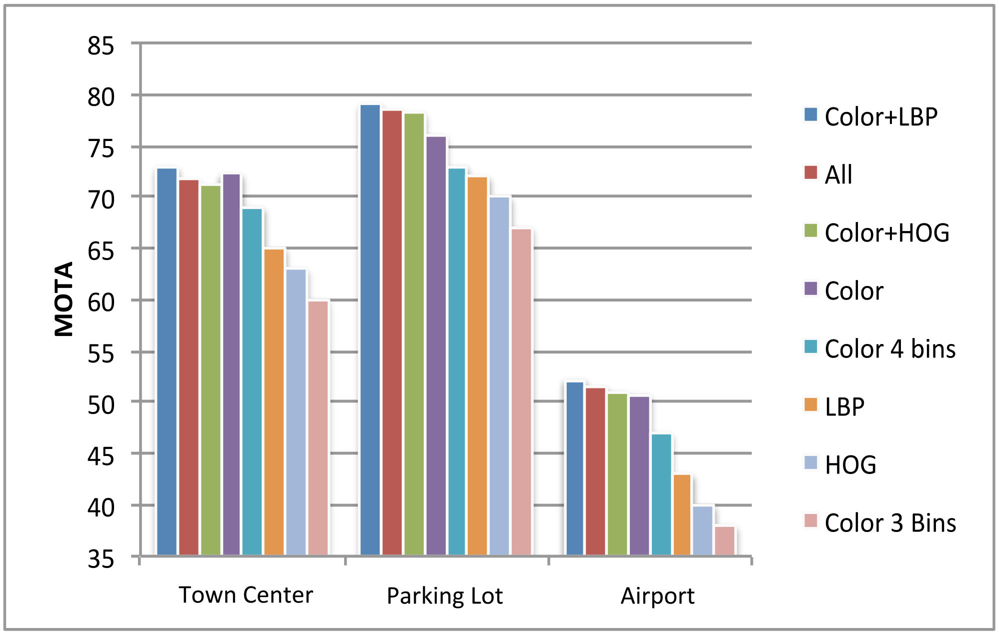


Figure 4.9: Tracking accuracy (MOTA) using different feature combinations. Four features are tested: LBP, HOG and color histogram. We built the 3D color-histogram features with 3, 4 and 5 bins, receptively. The experimental result shows that the combination of color histogram with 5 bins and LBP is evidently the most distinguishing feature set for all the datasets.

We analyzed the performance of our approach using different feature combinations. Figure 4.9demonstrates the obtained results. The experimental result shows that the combination of color histogram with 5 bins and LBP is evidently the most distinguishing feature set for all the datasets. Figure 4.10, 4.11 and 4.12 shows example frames from three datasets with the tracking results overlaid. We also compared the proposed detection method with the original DPM detector. The qualitative result is shown in Figure 4.13. Finally we show an extremely difficult video clip in Figure 4.14, where our method successfully tracks multiple humans during heavy occlusions.

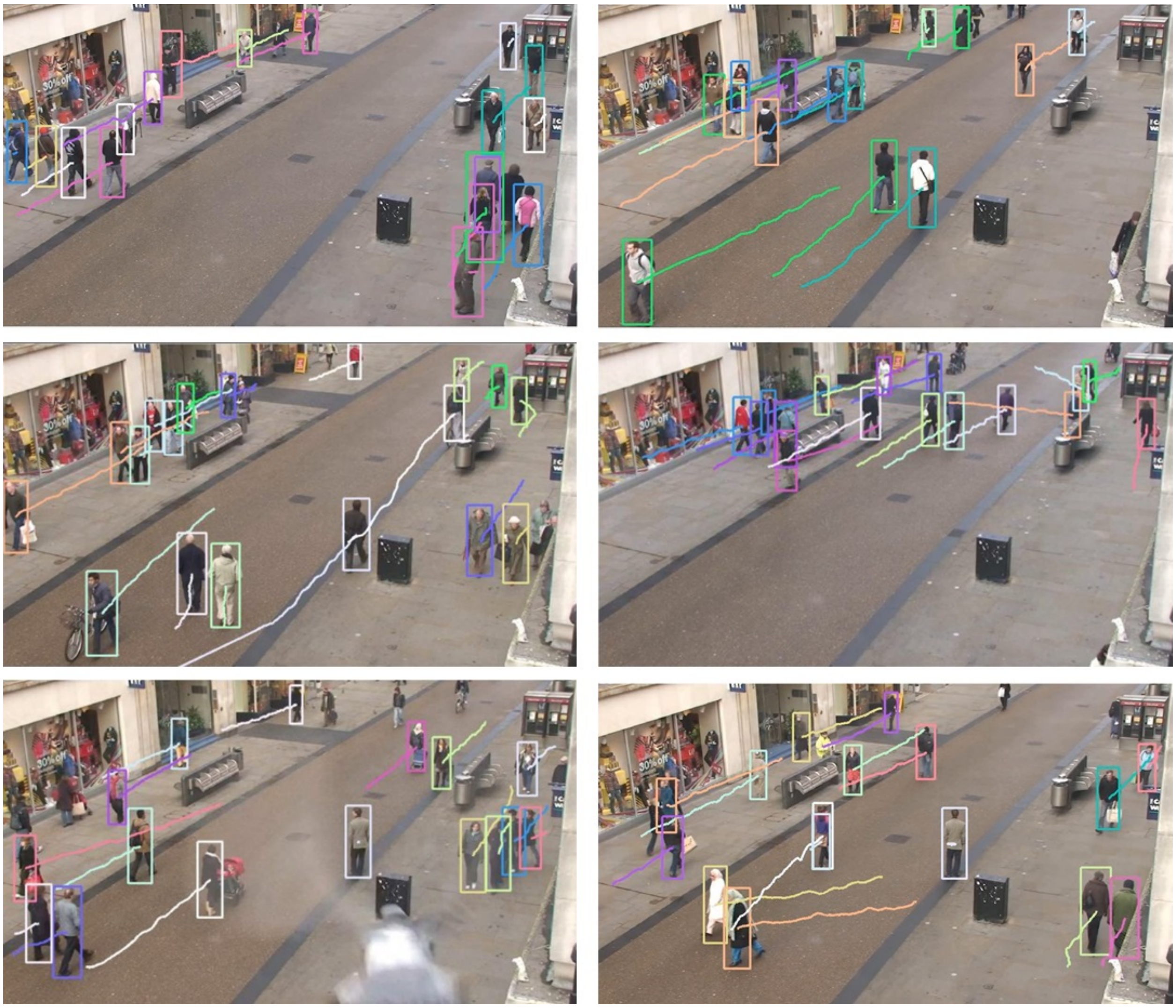


Figure 4.10: Tracking results of the Town Center sequence. Each image shows the tracking result up to that frame. The lines in different colors show the trajectories. Each individual is shown in different colors.

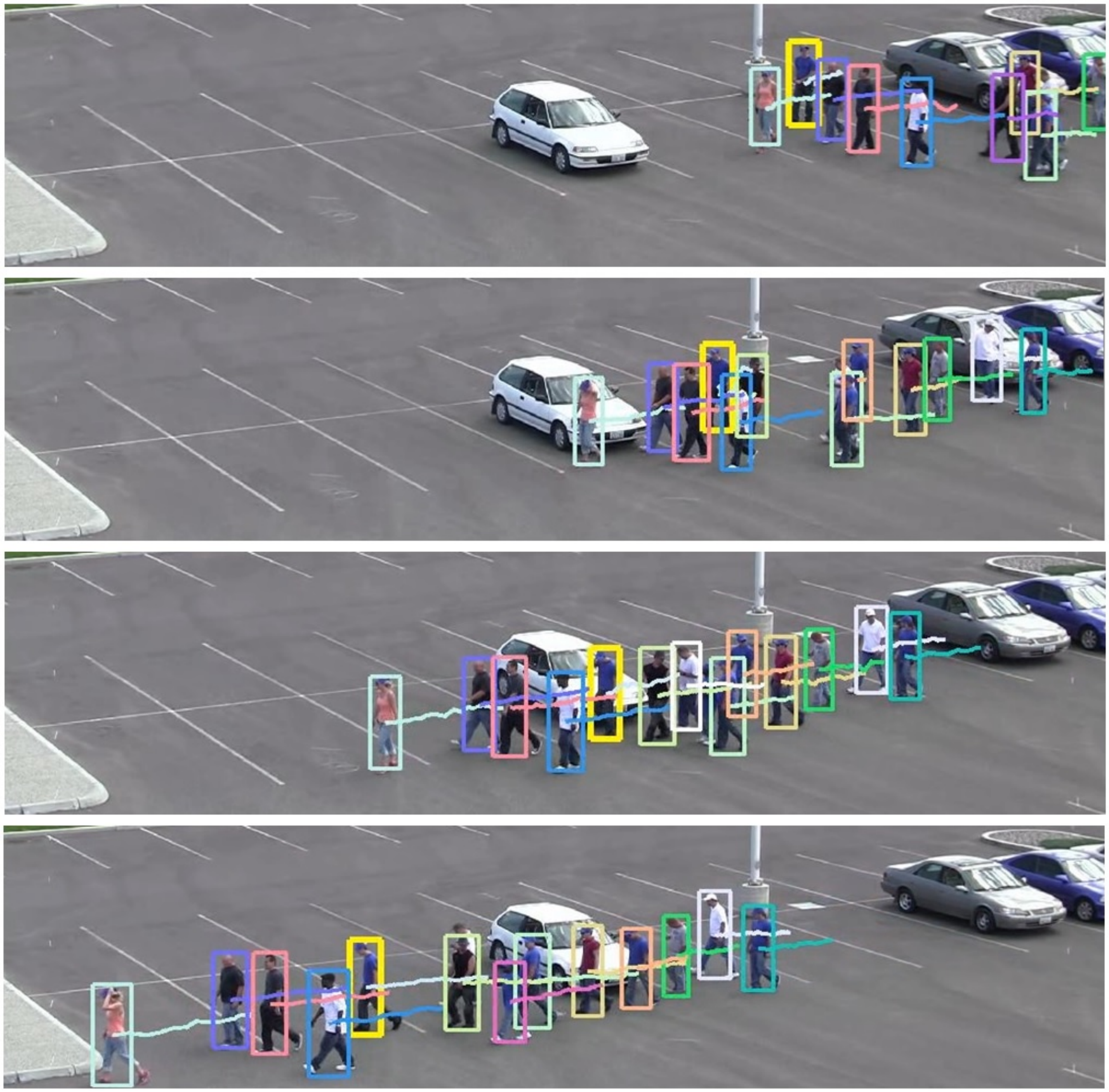


Figure 4.11: Tracking Results of the Parking Lot 1 sequence. Each image shows the tracking result up to that frame. The lines in different colors show the trajectories. Each individual is shown in different colors.

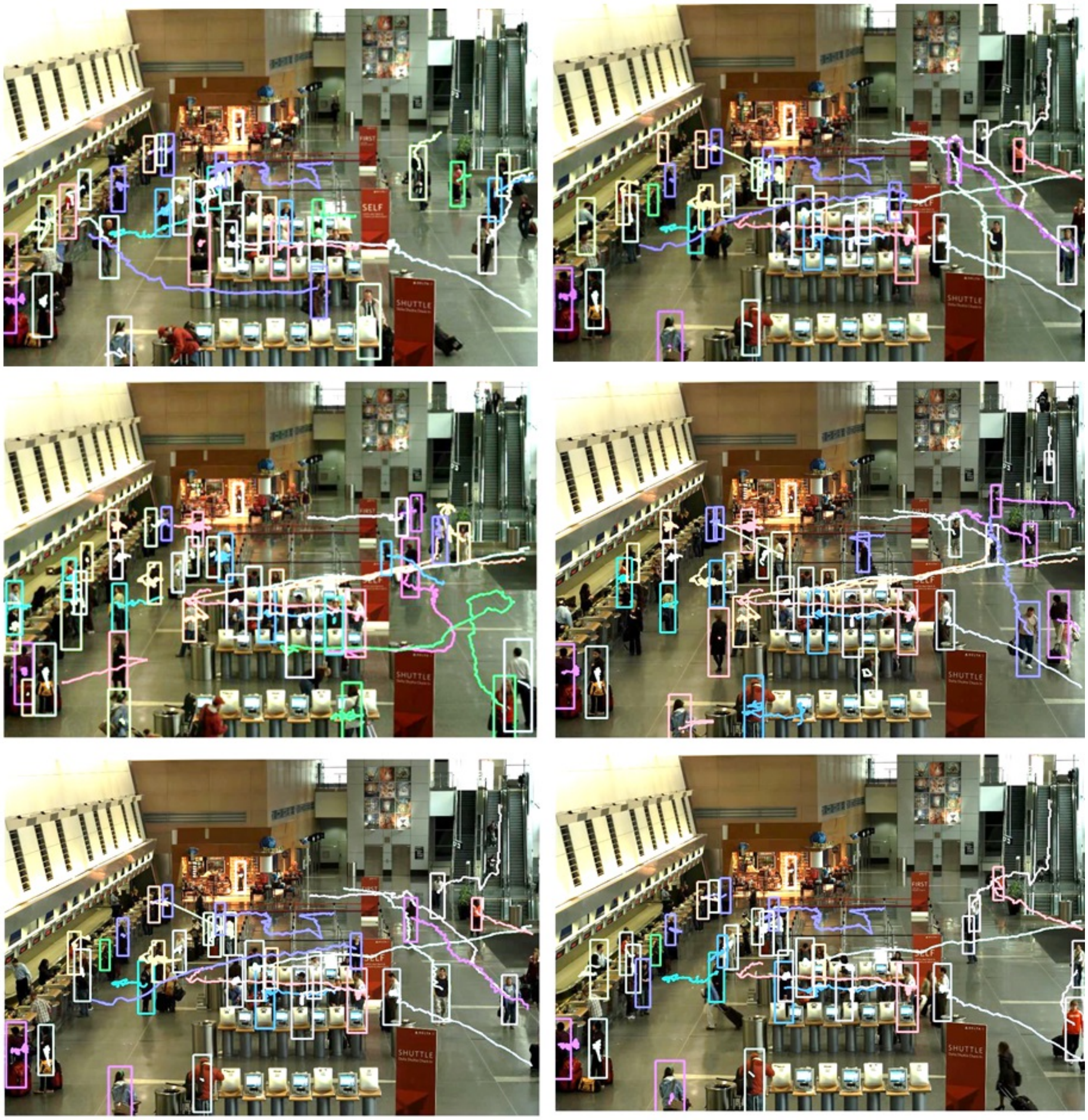


Figure 4.12: Tracking results of Airport sequence. Each image shows the tracking result up to that frame. The lines in different colors show the trajectories. Each individual is shown in different colors.

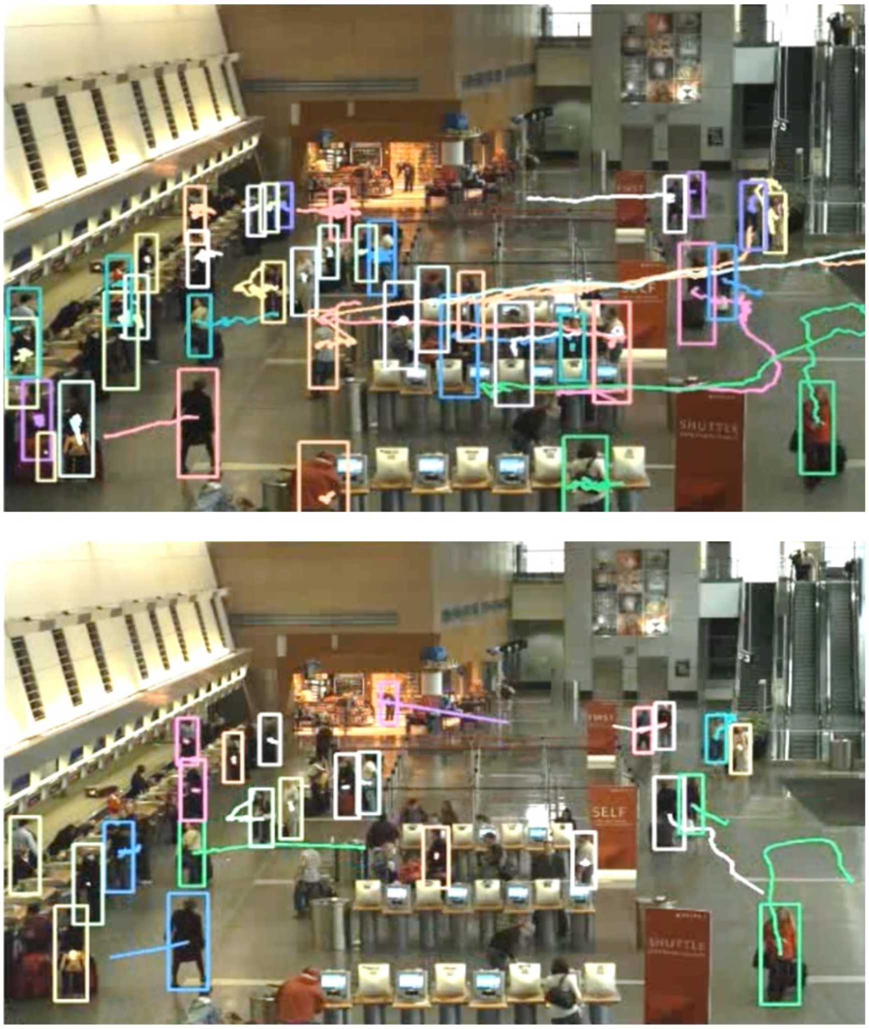


Figure 4.13: We compare the proposed detection method with the original DPM detector. The top figure shows the tracking result using the proposed method, while the bottom figure shows the result using the original DPM. It is clear that the proposed method obtains significantly more true detections and achieves better tracking performance.

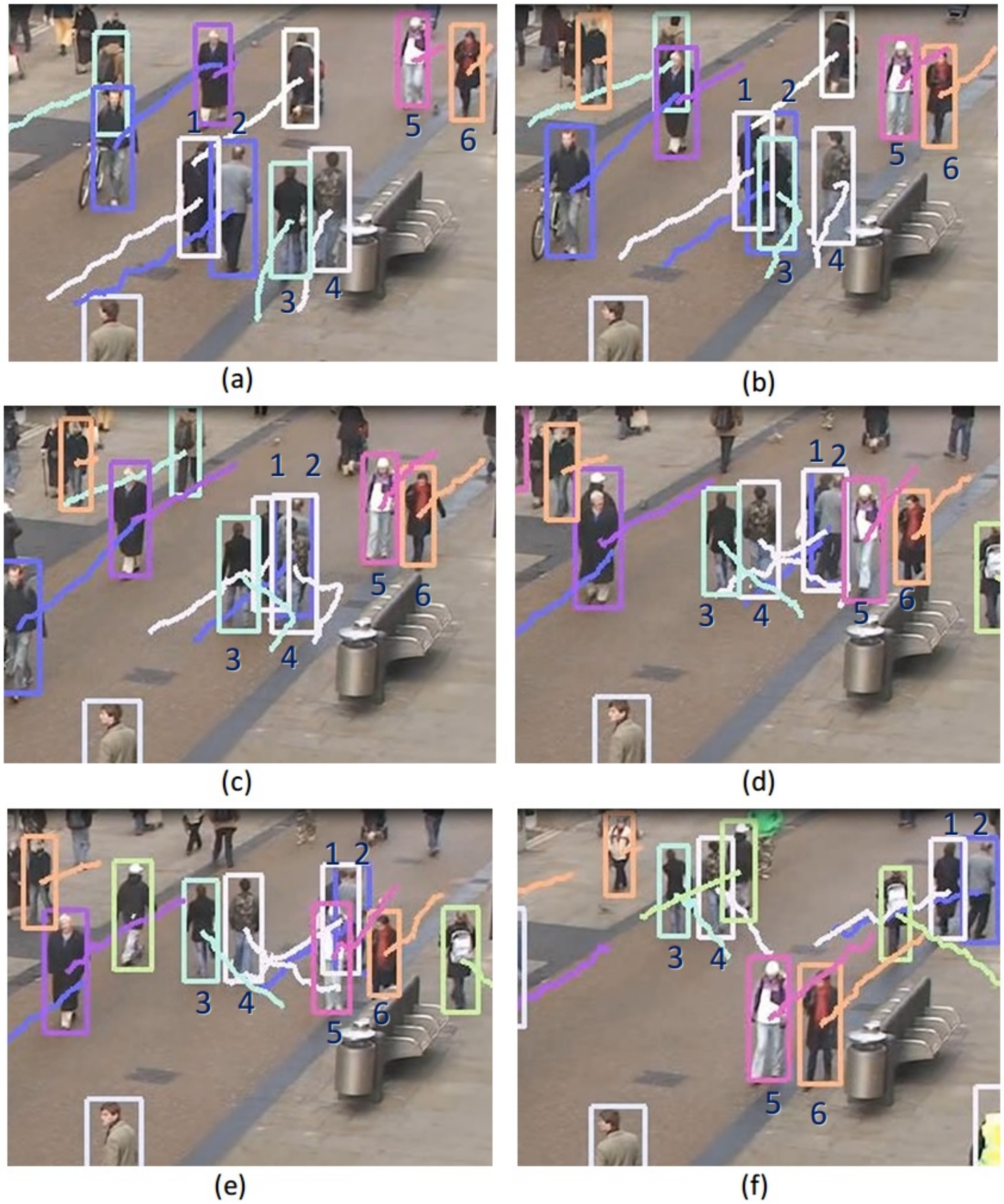


Figure 4.14: Figures (a) to (f) show our tracker successfully track multiple humans during extreme occlusions. The numbers in the figures show the IDs of different persons being tracked

4.6 Summary

In this chapter, we proposed an effective multiple-person tracking method using a part-based model and occlusion handling. Our method captures rich information about individuals; thus, it is highly discriminative and robust against appearance changes and occlusions. We employ an extended part-based human detector to obtain human part detections. Consequently, distinguishing person-specific classifiers are trained using the features of parts and then employed to associate the detections for tracking. We handle partial occlusions through dynamic occlusion reasoning and prediction across frames. We demonstrated by experiments that our tracking method outperforms state-of-the-art approaches in cluttered scenes.

CHAPTER 5: MULTIPLE HUMAN SEGMENTATION LEVERAGING HUMAN DETECTOR

In chapter 3 we presented a human detection method which gives superior results compared to a generic human detector. This template-based detector uses a sliding-window approach to go through the entire image and the detection output are represented as rectangle bounding boxes Therefore, the output of human detection is a bounding box covering a part of background and foreground with no clear separation of human region from the background. Instead of bounding boxes, in some applications such as person recognition, pose estimation, action recognition, and human tracking precise human silhouettes are desirable. In this chapter, we propose to explore a novel multiple human segmentation method which will leverage the output of human detector. The advantages of human segmentation are two folds: first, it can remove the irrelevant background pixels and provide clean object regions. Second, based on the segmentation, we can obtain more precise bounding boxes than the very rough original output. Recent works [8, 32, 78, 79] demonstrated that object segmentation can significantly improve the performance in detection or recognition tasks.

The rest of the chapter is organized as follows: The proposed framework is described in Section 5.1 with human detection in Section 5.1.1; initial segmentation at superpixel-level in Section 5.1.2; and multi-frame optimization in Section 5.1.3. Experimental results are provided in section 5.2.

5.1 Detection-Based Segmentation

The block diagram of our framework for multiple human segmentation in a video is shown in figure 5.1. First, we obtain human detections from the video using the Deformable part-based detector [44]. Next, we use a two-stage coarse-to-fine strategy for segmentation. The first stage is to perform segmentation based on superpixels to obtain an estimated segmentation quickly. In this stage we build a background model based on superpixels, and leverage the part-based detection potentials. The second stage is to refine the initial segmentation at the pixel-level, and generate the final output. In this stage we build tracklet-based GMMs and perform multi-frame CRF optimization. A detailed explanation of each step is provided in the rest of this section.

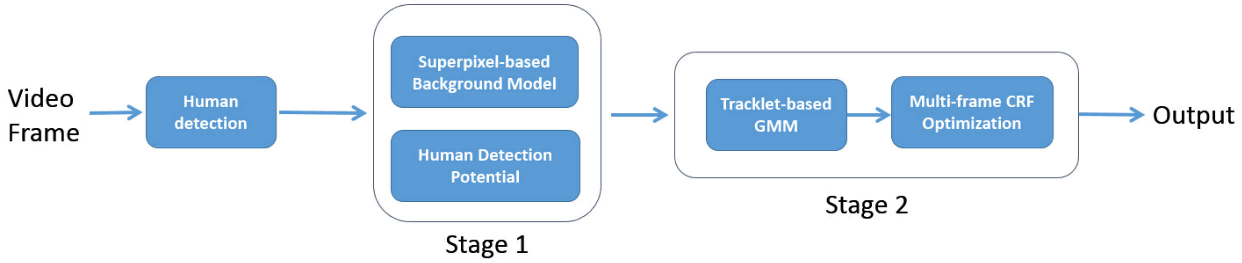

Video Frame
Human detection
Superpixel-based Background Model
Human Detection Potential
Stage 1
Tracklet-based GMM
Multi-frame CRF Optimization
Stage 2
Output

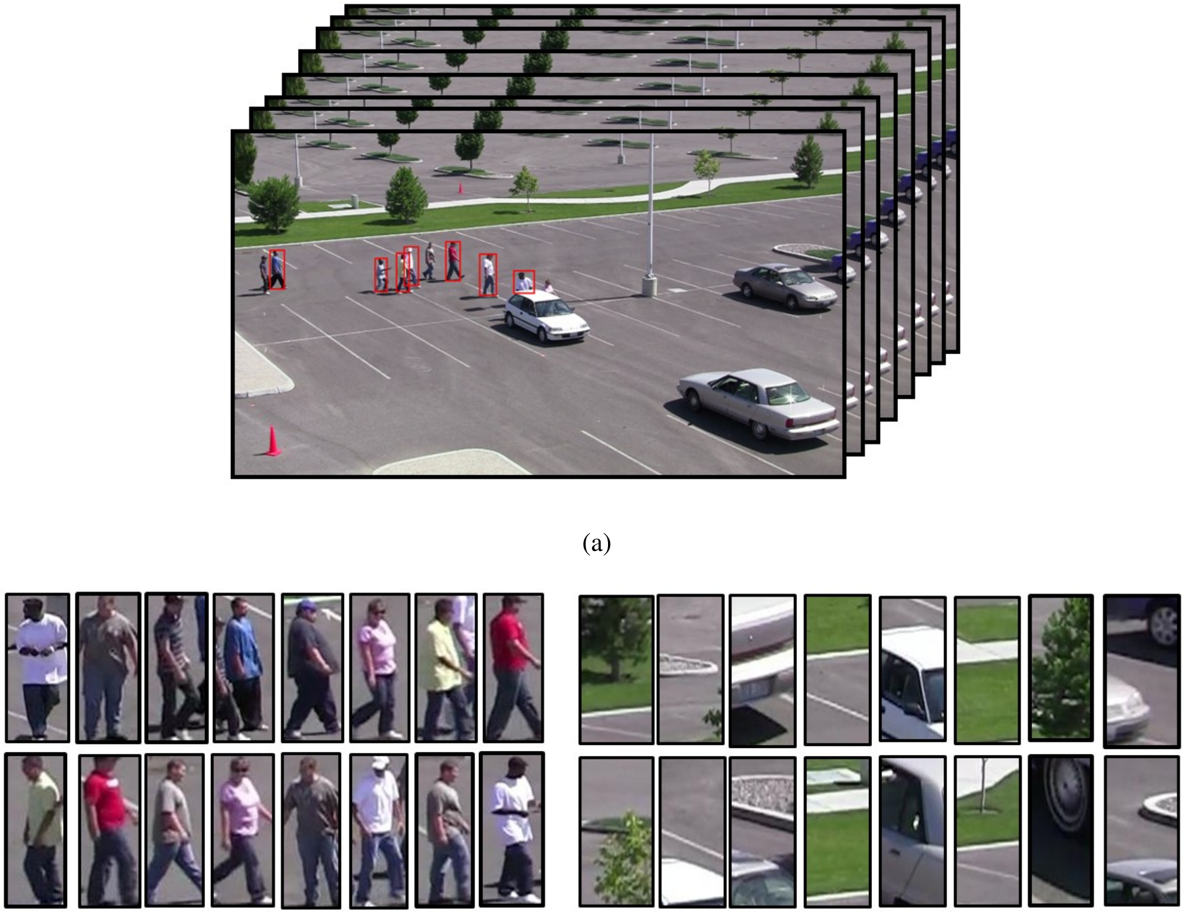
Figure 5.1: Block diagram of the proposed method. We use a two-stage coarse-to-fine strategy for segmentation.

5.1.1 Human Detection

At first, we employ a human detector to locate humans in a video frame, in the form of bounding boxes, which significantly reduces the searching space for the segmentation. Current human detectors have already achieved sufficient detection accuracy for ordinary images. We employ the deformable part-based model detector [43] as the initial detector in our approach since it has shown excellent performance of detecting pedestrians. Most importantly, it detects the essential parts of the human body, and these part potentials can be used to estimate the foreground distribution.

The DPM detector is trained using the PASCAL VOC 2011 training dataset [41], which consists of hundreds of manually annotated human examples. We further augment the detector with an unsupervised learning framework [93], which improves the detection accuracy by adapting the pre-trained detector in a specific video.

We run the human detector over each frame of the video, and collect a large number of detection examples $\mathbf{Dt} = \{d_i \mid i = 1, 2, ..., n\}$. The detector is given a lower detection threshold (high recall) so we can obtain almost all true detections and a large number of false detections, which will be used as background examples.



(a)

(b)

(c)

Figure 5.2: Detection results from a DPM detector. (a): detection output on a video sequence. (b): Examples from positive detection set $\mathbf{D_P}$. (c): Examples from negative detection set $\mathbf{D_N}$.

The detector returns a confidence score associated with each $d_i \in \mathbf{Dt}$, which we denote as $c(d_i)$. According to the confidence scores, we initially select two sets $\langle \mathbf{D_P}, \mathbf{D_N} \rangle$, where

$$\mathbf{D_P} = \{d_i \mid d_i \in \mathbf{Dt}, c(d_i) > t_p\}\,, \tag{5.1}$$

$$\mathbf{D_N} = \{d_i \mid d_i \in \mathbf{Dt}, c(d_i) < t_n\}\,. \tag{5.2}$$

$\mathbf{D_P}$ is used to initialize the segmentation; $\mathbf{D_N}$ is used to learn a background model in section 3.2. Figure 5.2 shows the detection results.

5.1.2 Initial Segmentation at Superpixel-Level

For the first stage, we develop an algorithm that takes the detection bounding box as the input and generate a binary segmentation as the output. The details are given below.

5.1.2.1 Superpixel Segmentation and GMM

Firstly, we perform superpixel segmentation of the positive set $\mathbf{D_P}$ and the negative set $\mathbf{D_N}$. A superpixel is a small aggregation of connected pixels with similar color or intensity. As the middle-level feature, superpixels enable us to measure the feature statistics on a semantically meaningful sub-region rather than individual pixels which can be brittle. To segment superpixels, we employ the simple linear iterative clustering (SLIC) method [4]. We segment each detection into n_{sp} superpixels. We choose an appropriate number for n_{sp} so that each superpixel is roughly uniform in color and naturally preserves the boundaries of objects. In order to encode color information into superpixels, we describe each superpixel Sp_i by a 3-dimensional feature vector $f = (L, a, b)$, in which (L, a, b) is the average $CIELAB$ colorspace value of all pixels. Figure 5.3(b) shows an example of the superpixel segmentation.

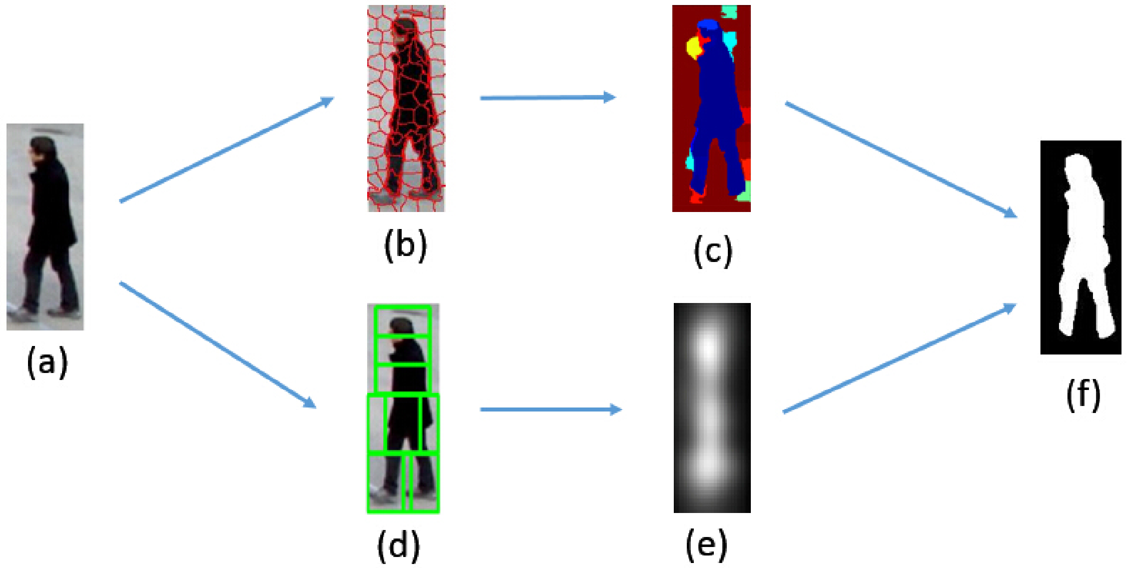


Figure 5.3: (a) Human detection bounding box. (b): Superpixel segmentation. (c): background confidence map. The red regions are with high confidence; the blue regions are with low confidence (d): Detected parts. (e): Detection potential. Brighter pixels show high confidence (f): Initial segmentation

Next, we build a background model in terms of GMM at superpixel-level for the scene using all the examples in the negative set $\mathbf{D_N}$. Consider a Gaussian mixture model with $M > 1$ components.

$$p(x|\omega) = \sum_{m=1}^{M} a_m p(x|\omega_m), \tag{5.3}$$

where a_m is the mixing proportions, each ω_m is the set of parameters defining the m-th Gaussian component. Let P_b^i denotes the probability of the i-th superpixel that belongs to the background, we can obtain a background confidence map P_b for each detection using equation 5.3, as shown in 5.3(c).

5.1.2.2 Part-Based Detection Potential

Unfortunately, the color-based GMM is not always reliable, especially when the appearances of the pedestrians are very similar to the background. In this case, a shape-based prior can provide additional information to the foreground distribution. Each human instance in the video has a specific part structure visible in the image due to the body articulation and viewpoint. Instead of using a generative shape prior, we leverage the part-based DPM detector, which optimizes the positions of the parts for each human instance so that they can best fit that instance.

We obtain the detection potential P_d using the response of all the parts. Let H denotes the Histogram of Oriental Gradients (HOG) feature of the detection window, and $p = (x, y)$ denotes a part specified by its position. P_d is computed from all the parts and the root filters considering the deformable cost

$$P_d = exp(-b(w_r p_0 + \sum_{i=1}^{n_p} s(p_i))^{-1}), \tag{5.4}$$

where p_0 is the response of the root filter, w_r is the weight of the root filter, b is a constant parameter, n_p is the number of parts, and $s(p_i)$ is the response of part i which is computed as

$$s(p_i) = F_{p_i} \cdot \phi(H, p_i) - d_{p_i} \cdot \phi_d(d_x, d_y), \tag{5.5}$$

where F_{p_i} is the part filter, $\phi(H, p_i)$ denotes the vector obtained by concatenating the feature vectors from H corresponding to the sub-window of the part p_i. (d_x, d_y) is the displacement of the part with respect to its anchor position, $\phi_d(d_x, d_y) = (d_x, d_y, d_x^2, d_y^2)$ represents the deformation features, and d_{p_i} specifies the coefficients of the deformation features.

Figure 5.3(d) shows an example of calculating detection potential. The first row shows the input image, the DPM detection, the root potential and the final detection potential; the second row shows the responses for all eight parts. The part responses show high density on head, torso, arms and legs separately.

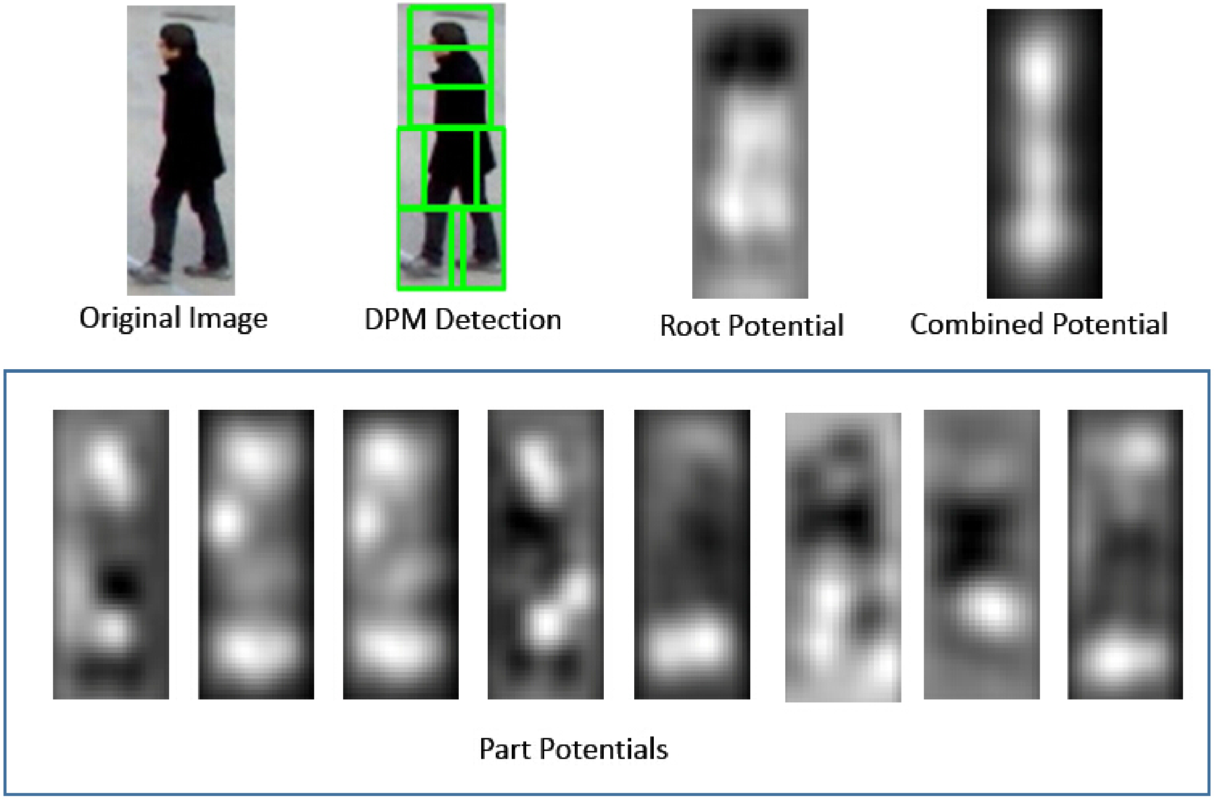


Figure 5.4: An example for calculating detection potential. The first row shows the input image, part-based detection, root filter potential and the combined potential; the second row shows the responses for all eight part filters.

5.1.2.3 CRF Optimization

In order to extract a precise region in the confidence map, a conditional random field (CRF) model [10] is utilized to learn the conditional distribution over the class labeling. CRF allows us to incorporate constraints in the pairwise edge potentials and hence improve the classification accuracy around the boundary. We define the energy function for labeling all the superpixels:

$$E(f) = \sum_{i \in I_{sp}} D(i, f_i) + \beta \sum_{(i,j) \in N_{sp}} V(i, j, f_i, f_j), \tag{5.6}$$

where I_{sp} is the set of all superpixel indices, N_{sp} is the set of 8-connected adjacent superpixel indices. $f = [f_1, f_2, ..., f_n]$ is the set of all superpixel labels, f_i could be set to 0 or 1 which stands for background or foreground respectively. The unary term $D(i, f_i)$ defines the cost of labeling superpixel i with label f_i by

$$D(i, f_i) = \begin{cases} -log(P_d(i)), & \text{if } f_i = 1 \\ -log(P_b(i)), & \text{if } f_i = 0 \end{cases} \tag{5.7}$$

where P_b is the background confidence map and P_d is the detection potential.

The binary term $V(i, j, f_i, f_j)$ is defined by

$$V(i, j, f_i, f_j) = [f_i \neq f_j] exp^{-\beta \|Sp_i - Sp_j\|}, \tag{5.8}$$

where [.] is the one-zero indicator function, $\|Sp_i - Sp_j\|$ is the $L2$-norm of color difference between adjacent superpixels, and $\beta = (2 \sum \|Sp_i - Sp_j\|)^{-1}|_{(i,j) \in N_{sp}}$ is the normalization term.

The CRF formulation can be optimized using the graph-cut algorithm [19]. Then we will obtain a binary segmentation map where the target and background are distinctly separated, as shown in Figure 5.3(e). We show a few more segmentation examples in Figure 5.5.

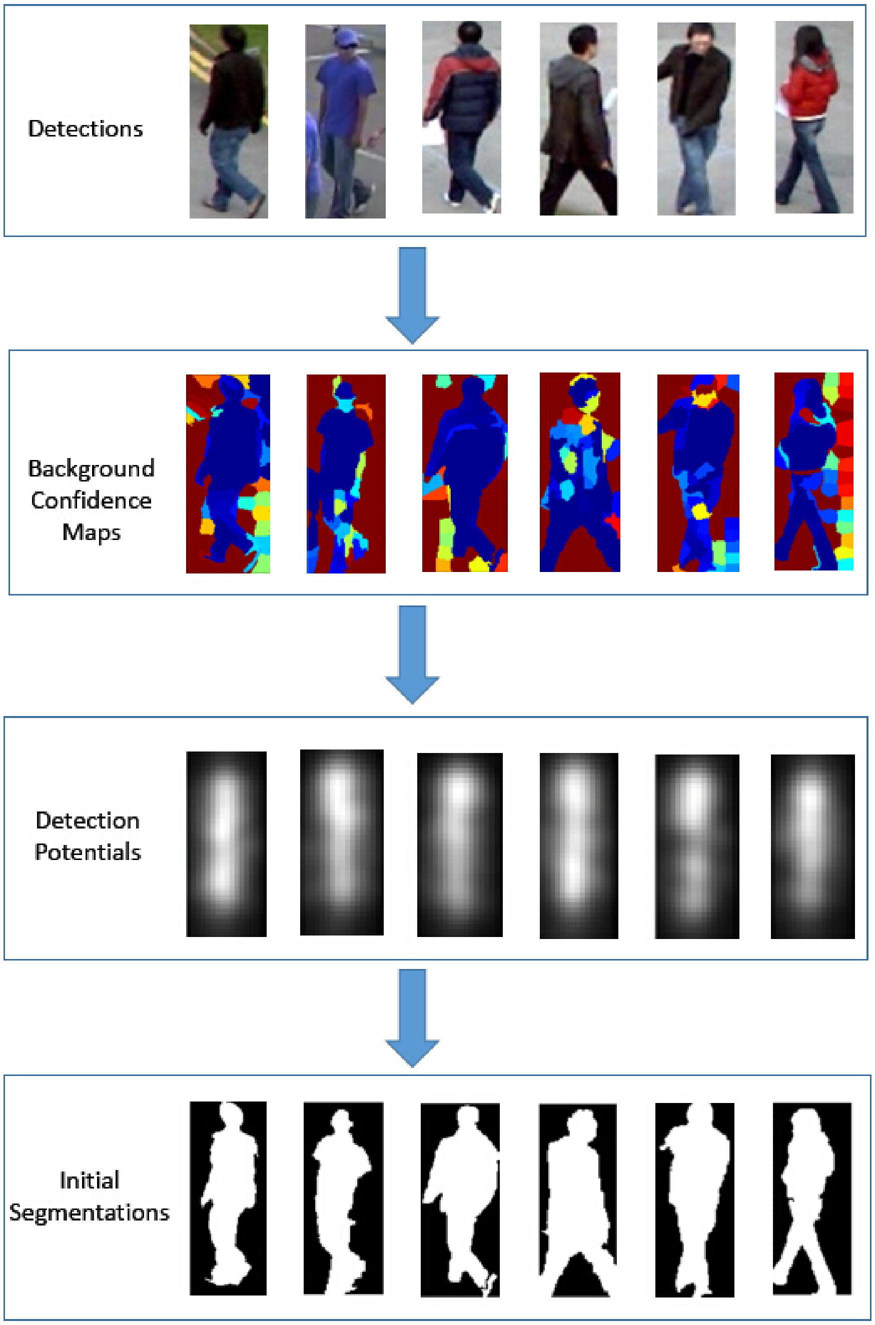


Figure 5.5: Segmentation results at superpixel-level. From the first row to the fourth row: human detections, background confidence maps, Detection potentials and initial binary segmentations.

5.1.3 Multi-Frame Optimization at Pixel-Level

The initial segmentation at the superpixel-level provides a good estimation for each detection window. However, image details are typically ignored by the superpixels. In order to obtain a more precise segmentation, we need to perform a second level segmentation at the pixel-level. Since we already have the initial segmentation, we now have a better estimation for the foreground and background areas. We build the pixel-level background and foreground GMMs, and compute probabilities.

5.1.3.1 Tracklet-Based GMM

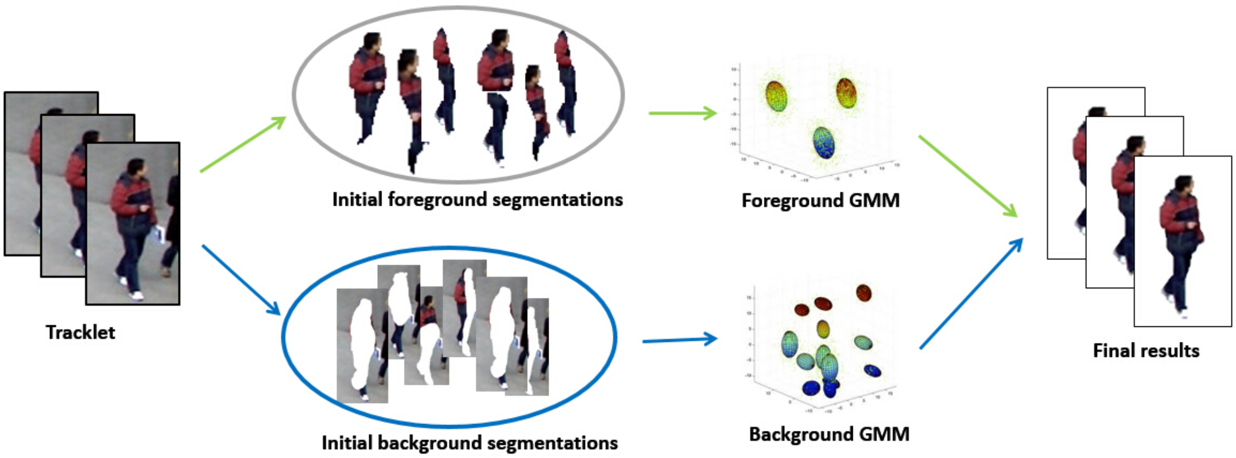

Initial foreground segmentations
Foreground GMM
Tracklet
Final results
Background GMM
Initial background segmentations

Figure 5.6: Building tracklet-based GMMs for foreground and background.

Although reasonable, the GMM based on a single bounding box does not take advantage of temporal continuity in the video. An object that appears in one frame of a video is certain to appear spatially close in neighboring frames as well. Therefore, instead of using only one detection window in a single frame, we build a multi-frame GMM based on a tracklet which consists of a few spatio-temporal consistent detections for the same person. A tracklet can provide more appearance and motion constraints which helps to reduce noise in any single detection window. Moreover, a tracklet can be short, typically about 15 frames, as shown in figure 5.7. These tracklets are obtained by associating the detections in different frames, which is described in section 5.1.3.2. Figure 5.6 shows steps involved in building a multi-frame GMM. In this figure, the detections from a tracklet are initially segmented into foregrounds and backgrounds, which are used to build the foreground/background GMMs. Although the initial segmentation for each detection in a single frame is not accurate, together they are used to build a more reliable GMM, which results in more accurate segmentation.

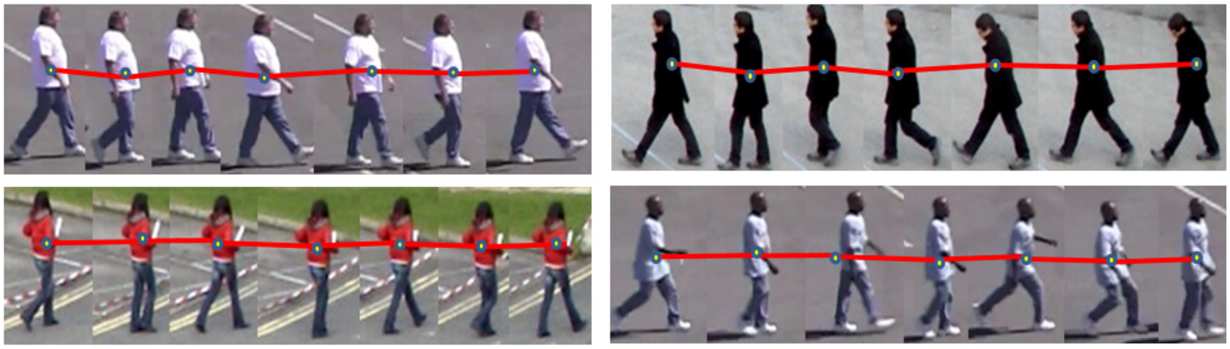


Figure 5.7: Examples of tracklets. Each subfigure shows a person being tracked over a short time. The trajectories are shown in red.

5.1.3.2 Obtaining Tracklets

In order to build a more reliable GMM for each person, the initial human segmentations in all the frames are grouped into small tracklets by their similarities, which depend on the position and color information between each pair of detections in adjacent frames.

The similarity between each pair of segmentations is computed as:

$$S_t = S_{hist}(r_m, r_n) \cdot S_{ovlp}(r_m, r_n), \tag{5.9}$$

in which S_{hist} is the color histogram similarity in $CIELAB$ space, and S_{ovlp} is the overlapping ratio between two regions measuring the location similarity, which is computed as:

$$S_{ovlp}(r_m, r_n) = \frac{r_m \cap r_n}{r_m \cup r_n}. \tag{5.10}$$

A threshold is set to decide whether two segments should belong to the same person, and a global data association scheme is employed to connect the initial segments into several tracklets which may be broken at any frame when the similarity is too low.

The tracklets are used to build appearance model for the corresponding persons.

5.1.3.3 Multi-Layer Graph-Cut

For each tracklet in the video, we solve a multi-layer graph-cut problem to obtain the final per-pixel segmentation results. A spatio-temporal graph is defined by connecting the detection windows within a tracklet. Figure 5.8 illustrates the graph structure in the spatio-temporal domain. Each node in the graph correspond a pixel in a frame; each node is connected to 8 neighbours within current frame and the forward-backward 18 neighbours in the adjacent frames. The energy function is defined for labeling all the pixels in the multi-layer graph:

$$E(f) = \sum_{i \in I_{px}} D(i, f_i) + \lambda \sum_{(i,j) \in N_{3d}} V(i, j, f_i, f_j), \tag{5.11}$$

where I_{px} is the set of all pixel indices, N_{3d} is the set of 3-D adjacent pixel indices. $f = [f_1, f_2, ..., f_n]$ is the set of all pixel labels, f_i could be set to 0 or 1 which stands for background or foreground respectively. The unary term $D(i, f_i)$ defines the cost of labeling pixel i with label f_i which we obtain from the multi-frame GMMs:

$$D(i, f_i) = \begin{cases} -log(U_f(i)), & \text{if } f_i = 1 \\ -log(U_b(i)), & \text{if } f_i = 0 \end{cases} \tag{5.12}$$

where U_f is the foreground probability and U_b is the background probability. The foreground GMM is built using the initial human segments. The background GMM is built by removing the segments from the bounding boxes.

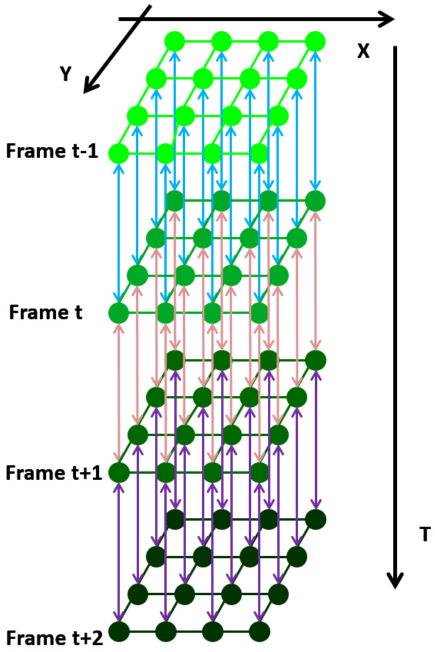


Figure 5.8: A spatio-temporal graph for segmentation. Each layer in the graph correspond a frame in the video; each node correspond a pixel in a frame.

The binary term $V(i, j, f_i, f_j)$ is defined by

$$V(i, j, f_i, f_j) = [f_i \neq f_j] exp^{-\beta \|c_i - c_j\|}, \tag{5.13}$$

where [.] denotes the indicator function taking values 0 and 1, $\|c_i - c_j\|$ is the $L2$-norm of color difference between two adjacent nodes, and $\beta = (2 \sum \|c_i - c_j\|)^{-1}|_{(i,j) \in N_{3d}}$ is the normalization term.

Algorithm 2 shows different steps of the proposed method. The two stages in our method are tightly connected and complement each other for human segmentation. As discussed earlier in this chapter, the superpixel-level segmentation can provide a quick and rough estimation for further optimization; On the other hand, the multi-frame pixel-level segmentation can make more precise boundaries, but it requires good initializations. It is a natural choice to combine the two levels and take advantages from both. The experimental results also show that by combining the two stages we can achieve the best overall performance.

Algorithm 2 Segmentation based on human detections and tracklets

```
Input:
    A video sequence $V$
Output:
    Segmentation Output, $O$;
Obtain initial positive detections $\mathbf{D_P}$ and negative detections $\mathbf{D_N}$ using the DPM detector;
Extract superpixels $S_p$ from $\mathbf{D_P}$ and $\mathbf{D_N}$;
Learn the background GMM based on $\mathbf{D_N}$ and compute background probability $P_b$;
Obtain detection potentials $P_d$ based on the DPM outputs;
Perform CRF optimization and obtain initial segmentations $O_0$;
Obtain tracklets $T_r$ based on $\mathbf{D_P}$;
Learn multi-frame GMMs based on $O_0$ and $T_r$;
Compute background/foreground probability based on the GMMs;
Perform multi-frame CRF optimization and obtain final output $O$;
return $O$;
```

5.2 Experimental Results

5.2.1 Evaluation Dataset

We experimented the proposed method extensively using 6 video sequences, which provide a wide range of significant challenges including illumination variations and cluttered backgrounds.

PNNL Parking Lot [92] (PL1 & PL2) We put together a high-resolution dataset which contains two video sequences of a parking lot using a static camera. These video sequences have 1,000 and 1,533 frames respectively, each of size 1920×1080. This dataset contains video sequences of moderately crowded scenes including groups of pedestrians walking in queues with parallel motion and similar appearance.

PETS2009 [3] (PT) This sequence has a resolution of 768×576. We select one video sequence $(ID : S2 - L1 - Time12 - 34 - View001)$ which has 795 frames as a test video. This is a video of a relatively sparse scene including a few people walking in random directions. We up-sampled the resolution of this video by 2 because the original low resolution is not suitable for the DPM detector.

Skateboarding [93](SB1 & SB2) We put together a Skateboarding dataset which contains two video sequences captured by a hand-held camera. The first video has 367 frames, each of size 1280×720, the second video has 769 frames, each of size 640×480. This is a very challenging dataset due to the camera motion and extremes changes in pose.

Town Center [14] (TC) This is a high-resolution video sequence consisting of 7501 frames, each of size 1920×1080. This is a video of a semi-crowded scene with rare long-term occlusions. The motion of pedestrians is often linear and predictable. However, it is still quite challenging because of the inter-object occlusions and the difficulty of detecting pedestrians on a bike or with a stroller.

5.2.2 Ground truth

We use the first 800 frames from each video sequences for our experiments. To quantitatively evaluate our method, we annotated the ground-truth for every 10 frames in these video sequences. We manually segmented all humans from the sub-sampled video frames. Figure 5.9 shows examples of the ground truth data from each dataset. The dataset with groundtruth will be made publicly available.

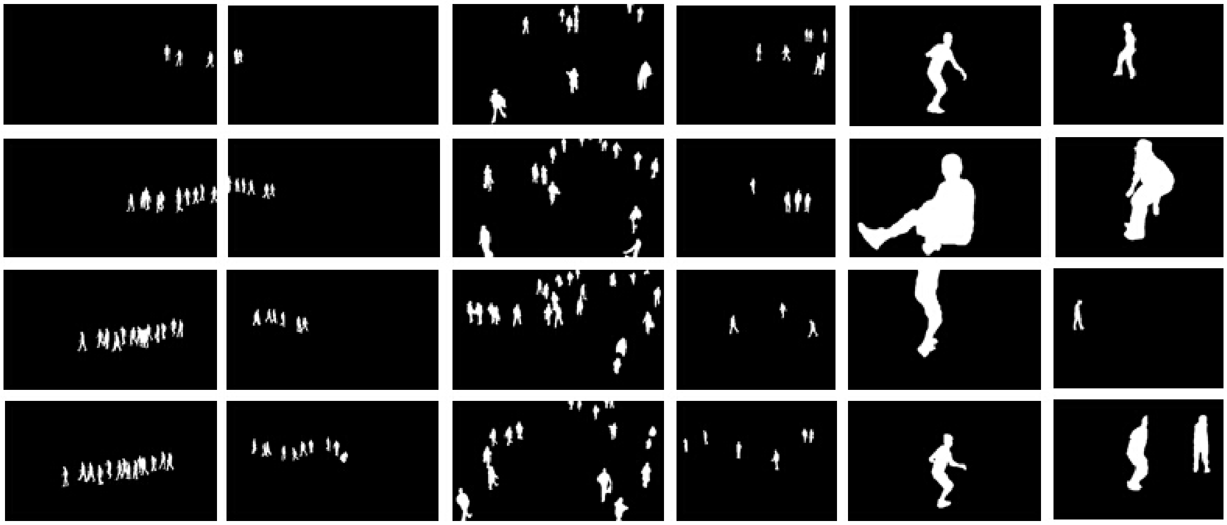


Figure 5.9: Example frames of the groundtruth data. Each column shows frames from different sequence. Foreground areas are shown in white.

5.2.3 Implementation Details

In our implementation, the DPM detector [44] is trained with the PASCAL2011 person dataset [42]. The DPM detection threshold is set to 0.5 to obtain positive detection set $\mathbf{D_P}$. For the superpixel segmentation and clustering, the number of superpixels in each detection bounding box

$n_{sp} = 100$, and the number of Gaussian $k = 10$. The maximum tracklet length is set to 15 for both efficiency and accuracy. We use version 5 of the DPM detector implementation [48] for all of our experiments.

5.2.4 Comparison with Baseline Methods

We compared the proposed approach with two baselines. Since our method is fully automatic, we compared it with the methods that require no human interaction. In addition, we evaluated the intermediate steps of the proposed method to know how they affect the segmentation accuracy.

Background Subtraction Background subtraction is a common approach for extracting motion regions in videos. This baseline runs a background subtraction method [53] on the whole video. No human detector is used with this baseline.

Graph-cut Segmentation In this baseline, we use the same DPM human detector to obtain detection bounding boxes, and employ the Graph-cut method [19] directly on these bounding boxes. The corners of bounding boxes are used for initializing the background areas.

Grab-cut Segmentation Similar to the Graph-cut baseline, we employ the Grab-cut method [88]. The corners of bounding boxes are used for initializing the background regions.

Ours-Stage 1 This is only the first stage of our method, which uses the background GMM and the detection potentials, and obtains the segmentation at the superpixel-level.

Ours-w/o tracklets This is the proposed method without using the temporal information. The GMM and the graph-cut in stage 2 are based on a single detection window.

Ours This is the proposed method including all the steps.

5.2.5 Evaluation Metric

We evaluated our segmentation results using the standard Segmentation Accuracy Metrics, which are used to measure and compare the performance of segmentation algorithms applied in

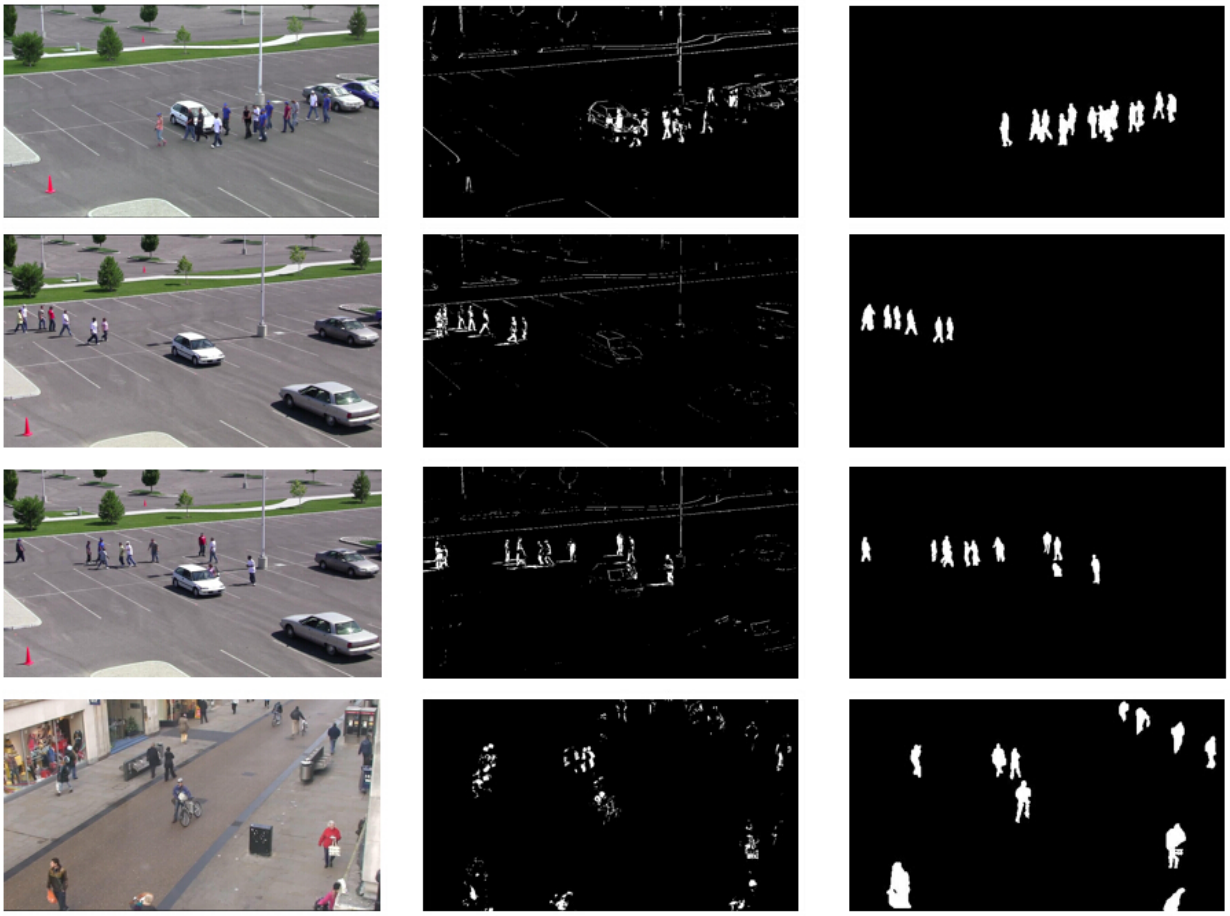


Figure 5.10: Comparison between background subtraction and our method. The left column shows the original video frames, the middle column shows the segmentation results by background subtraction, and the right column shows the segmentation results by our method

videos. In particular, we use Frame-based Segmentation Error and Identity-based Segmentation Accuracy. The Frame-based Segmentation Error is the average of per-pixel error over all frames in a video. It is defined as:

$$E_{frame} = \frac{1}{F} \cdot \sum_{i=1}^{F} |XOR(S_i, G_i)|, \tag{5.14}$$

where F is the number of frames in the video, S is the human segmentation results and G is the ground truth segmentation.

The Identity based Accuracy is the average of per-pixel intersection-over-union ([40]) score for all the people appearing in every frame. It is defined as:

$$A_{identity} = \frac{1}{\sum_{i=1}^{F}(Q_i)} \cdot \sum_{i=1}^{F}(\sum_{j=1}^{H_i} \frac{S_i^j \cap G_i^j}{S_i^j \cup G_i^j}), \tag{5.15}$$

where F is the number of frames, H_i is the number of people in frame i, S_i^j is the human segmentation results of the jth person in the ith frame, and G_i^j is the corresponding ground truth segmentation annotation.

5.2.6 Analysis of Results

We quantitatively compared our proposed method with three baseline methods, including background subtraction [53], Graph-cut [19] and Grab-cut [88]. We show the results in table 5.1 and 5.2 for 6 video sequences. Table 5.1 shows the segmentation performance measured by the Identity-based Accuracy, which evaluates the segmentation based on each detection window. The proposed method significantly outperforms the three baseline methods for all the tested datasets. It outperforms the Graph-cut by up to 18% and the Grab-cut by up to 13%, since we additionally leverage the color and shape information, as well as the spatio-temporal constraints from the videos.

Our method outperforms the background subtraction by 40% for the skateboarding dataset (SB1 and SB2). The reason is two-fold: firstly, background subtraction is sensitive to camera motion, and these videos have a considerable amount of camera motion; secondly, the motion from non-person objects, e.g. trees and shadows, also introduces a large amount of noise. In contrast, our method is based on human detection, which is not affected by camera motion and non-person objects. Figure 5.10 qualitatively compares the background subtraction and our method. We can see that the background subtraction results have a large amount of noise generated by the camera motion and object motion, while the results by our method are cleaner and more accurate.

Table 5.1: Identity-based Segmentation Accuracy

Sequence	PL1	PL2	PT	SB1	SB2	TC
BS [53]	0.61	0.62	0.65	0.35	0.21	0.53
Graph cut [19]	0.59	0.66	0.67	0.64	0.49	0.58
Grab cut [88]	0.64	0.69	0.73	0.69	0.51	0.60
Ours-stage 1	0.62	0.65	0.63	0.65	0.50	0.61
Ours(w/o tracklets)	0.66	0.71	0.76	0.72	0.51	0.65
Ours	**0.77**	**0.78**	**0.79**	**0.76**	**0.53**	**0.68**

Table 5.2: Frame-based Segmentation Error (lower numbers are better) $(\times 10^3)$

Sequence	PL1	PL2	PT	SB1	SB2	TC
BS [53]	2.00	2.62	3.17	58.6	9.16	6.18
Graphcut [19]	1.21	1.36	2.05	20.9	5.93	3.49
Grabcut [88]	1.11	1.22	1.85	17.4	5.60	3.34
Ours-stage1	1.16	1.41	2.50	20.5	5.79	3.31
Ours(w/o tracklets)	1.10	1.19	1.70	15.4	5.66	3.26
Ours	**0.82**	**0.95**	**1.55**	**13.2**	**5.59**	**3.03**

Furthermore, we compared the performance of intermediate steps of the proposed method. The 4^{th} row and the 5^{th} row show the results of our method with stage 1 only, our method without using the tracklets, respectively. We can observe that each step (superpixel-level segmentation, pixel-level segmentation and the multi-frame optimization) has considerable contribution to the final result. To compare the overall performance among all the methods, figure 5.11 shows the average segmentation accuracy over all the datasets for different methods; The proposed method outperforms all other methods significantly.

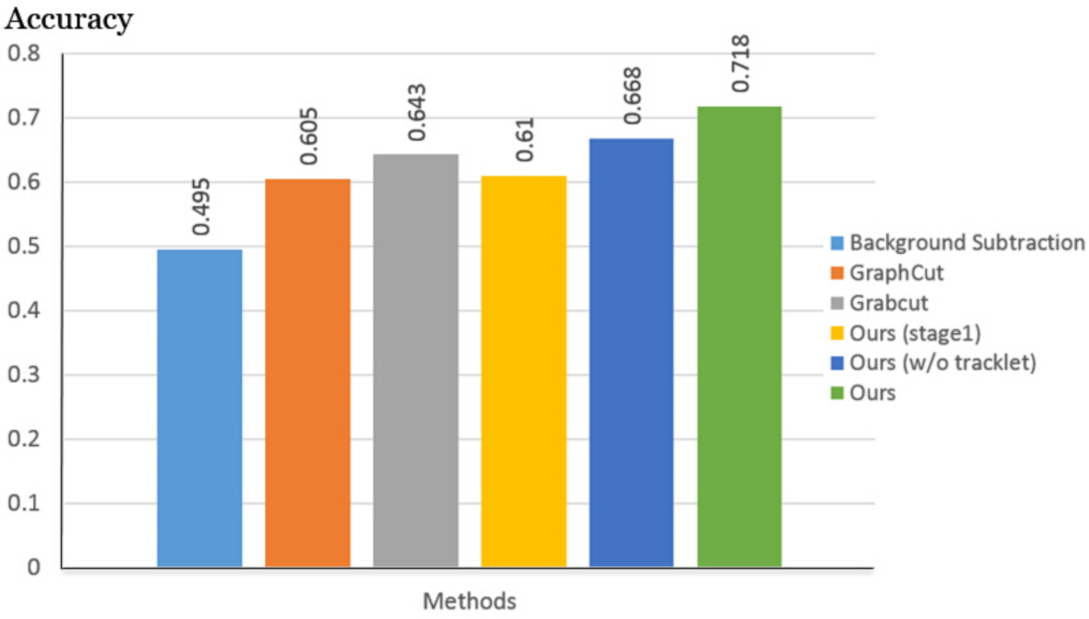


Figure 5.11: Average accuracy over all the datasets. Each method is marked by a different color

Table 5.2 shows the segmentation performance using the Frame-based Error metric, in which the lower numbers are better. The differences between table 5.1 and table 5.2 are: (a) Figure 5.2 measures the segmentation quality using the entire video frame, the background subtraction method in table 5.2 gets relatively low performance since the background noise is also considered. (b) A detection bounding box might contain a part of another person, which will be labeled as background using metric in table 5.1, but it will be labeled as foreground in table 5.2. Therefore table 5.2 is more suitable to evaluate the practical video segmentation tasks.

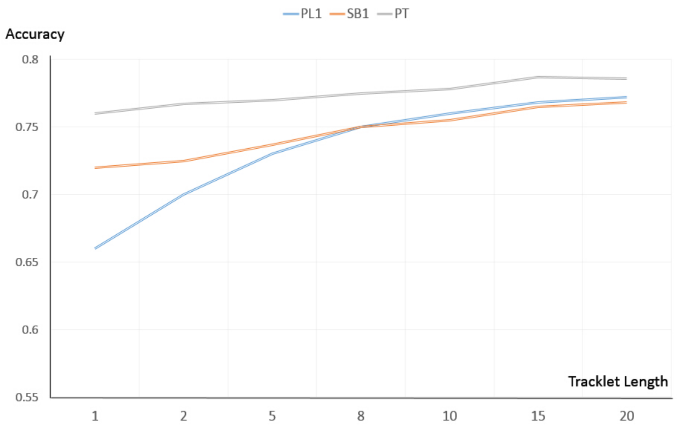


Figure 5.12: Accuracy as a function of tracklet length for PL1, PT and SB1 datasets.

We additionally evaluated how the segmentation performance is affected by different factors on three datasets: PL1, SB1 and PT. Figure 5.12 shows how the accuracy changes as a function of tracklet length. The tracklet length L is defined as the maximum number of frames in a tracklet; the actual tracklet length may be less than this value because of occlusion. We can observe that the segmentation accuracy increases as the tracklet length increases, and achieves the maximum when $L = 15$. When the length is 1, the accuracy is the lowest as no temporal information is used for segmentation. This shows that the tracklets and the multi-frame optimization improves the segmentation performance.

Figure 5.13 shows how accuracy changes as a function of the number of Gaussian k in GMM in stage 2. The segmentation accuracy increases as the number of Gaussian increases, and achieves the maximum when $k = 30$. Figure 5.14 shows the accuracy change with different number of superpixels in the stage 1. The segmentation accuracy increases as the number of superpixels increases, and achieves the maximum when $n = 150$. When the number is low, the superpixels are larger but with less precise boundaries, reducing the segmentation accuracy.

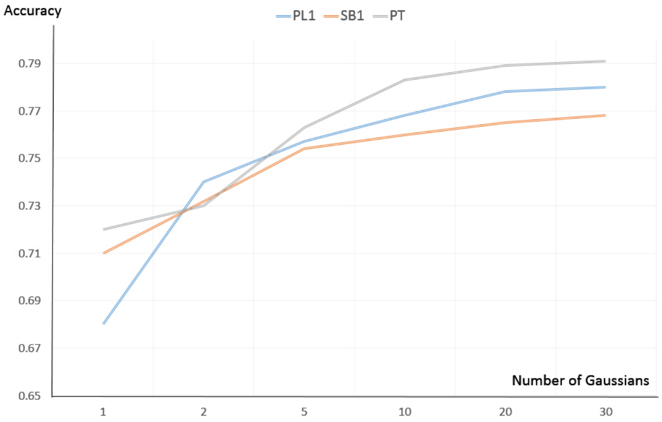


Figure 5.13: Accuracy for different number of Gaussians on PL1, PT and SB1 dataset

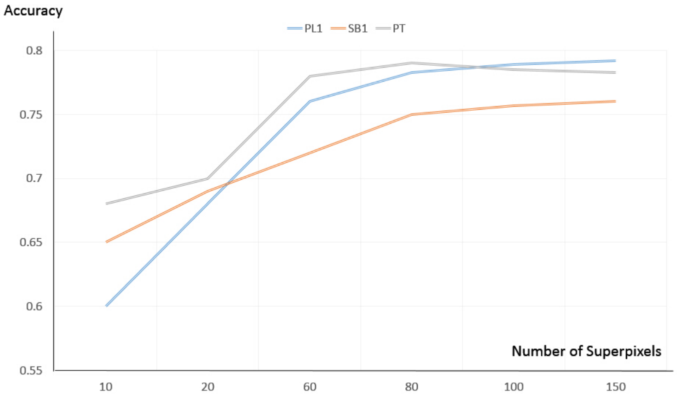


Figure 5.14: Accuracy for different numbers of superpixels on PL1, PT and SB1 dataset

To qualitatively evaluate our method, we further visualize the segmentation results. Figure 5.15 shows the segmentation results in entire video frames. The segmented region corresponding to human is shown in the original image in red. Figure 5.16 shows the results of the Grab-cut method and the intermediate segmentation results of the proposed method. From top to bottom they are; the original detection image; the results obtained using Grab-cut method; the proposed method with stage 1 only; the proposed method without using tracklets; and the proposed method. From the results we observed the segmentation is improved when more steps are applied: the human silhouette is more accurate and noise is decreased. Therefore, each step is indispensable to obtain the final results.

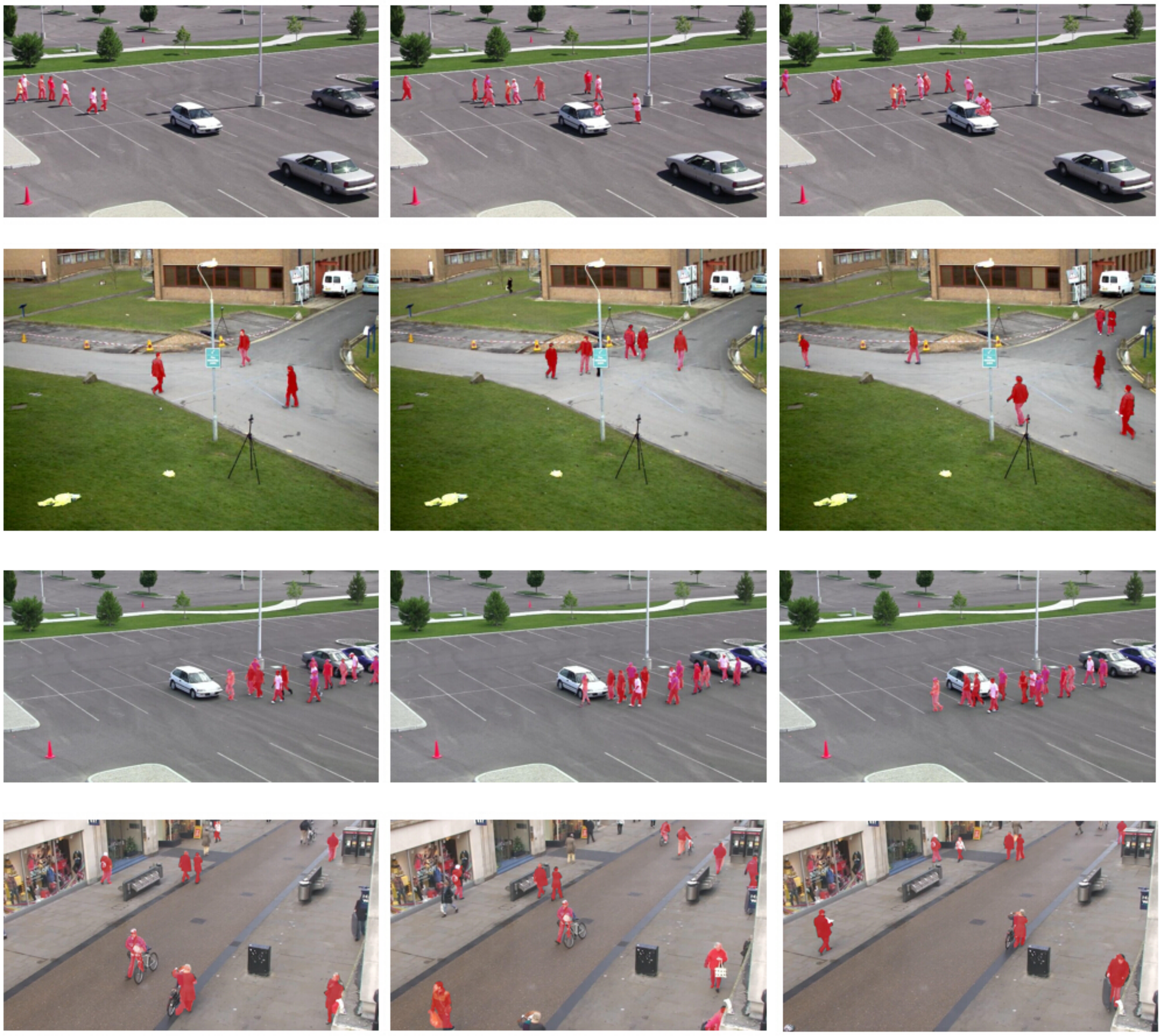


Figure 5.15: Human segmentation results for frame. From first row to the last row are video frames from PL2, PT and PL1 dataset, respectively. The segmented regions are shown in red.

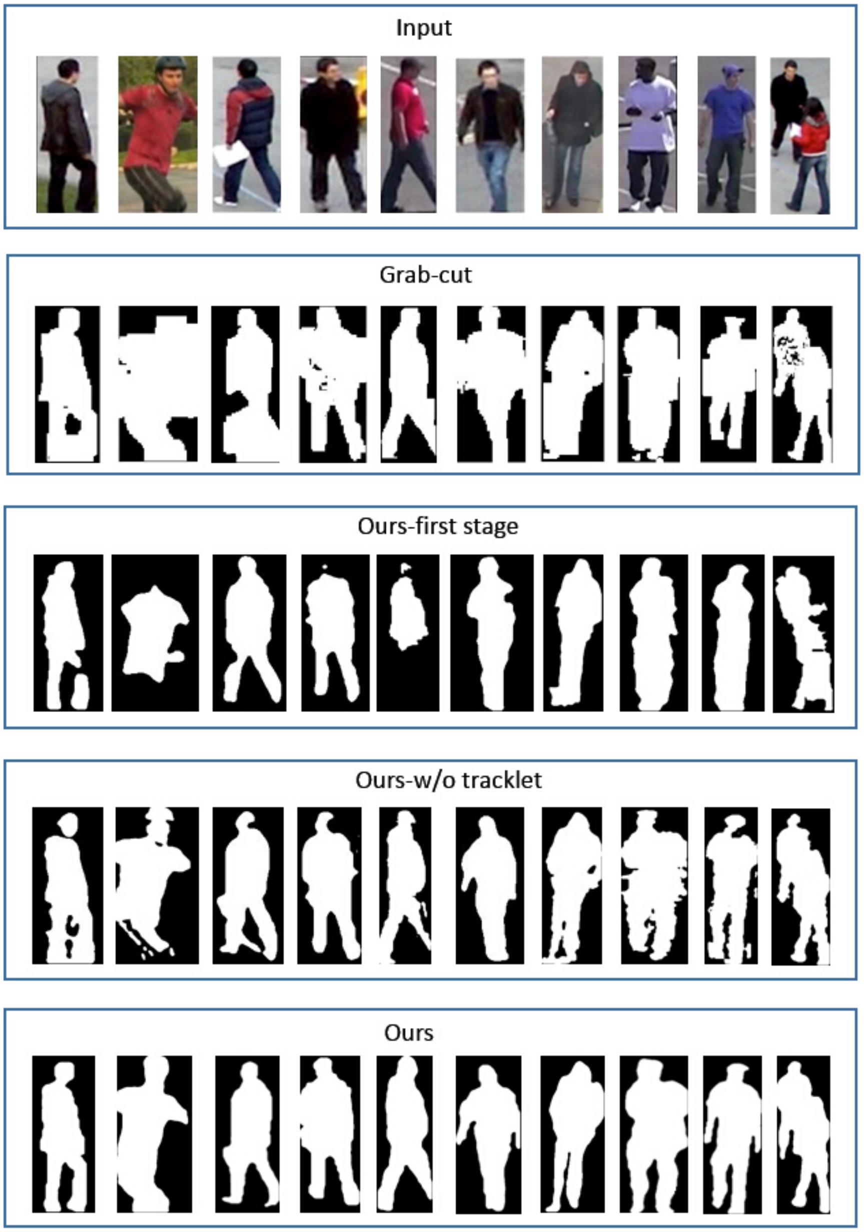


Figure 5.16: Qualitative segmentation results. Each column is a human instance and its segmentations using different methods. From top to bottom are image windows corresponding to humans, followed by results obtained using Grab-cut [88], using our method with stage 1 only, our method without using tracklets and the results obtained using our final proposed method, respectively.

5.3 Summary

In this chapter, we proposed an effective multiple human segmentation method based on DPM detection and multi-frame GMMs. Our method consists of two stages: Firstly it uses human detections as initialization and generates an initial segmentation at the superpixel-level. To refine the segmentation, we additionally leverage the part-based detection potentials provided by the DPM detector and perform finer segmentation at the pixel-level. Our method is based on the human detector, therefore it is fully automatic and is not sensitive to camera motion since it is does not use background difference. The experimental results show that our method significantly outperforms previous human segmentation methods.

CHAPTER 6: NONA: AN EFFICIENT TRACKING SYSTEM FOR HIGH DEFINITION VIDEOS

In the previous chapter, we proposed a robust multiple humans tracking method which leveraged a sophisticated part-based model. Although this approach achieves satisfying accuracy in realistic video data, its computational cost is relatively high, especially when it is applied to high-resolution videos. Since the latest generation of surveillance cameras can provide wide-area, mega-pixels videos, we desire a more efficient tracking system for a practical surveillance application.

In this chapter we will introduce an efficient tracking system named 'NONA', which is specially designed for tracking a single object in a high-definition video. It is implemented with a multi-thread architecture in order to deal with high computational cost. In addition, easy-to-calculate features and the FFT-based correlation algorithm are employed to further reduce the pressing time. The NONA system achieves real-time tracking performance in high-definition video while maintaining acceptable accuracy.

The rest of this chapter is organized as follows: In Section 5.1, we introduce the tracking environment and a new dataset. In Section 5.2, we describe different parts of the system including feature extraction, a motion model, template matching, adaptive template scaling and occlusion handling. In Section 5.3, we describe the multi-threading implementation details. Experimental results are presented in Section 5.4.

6.1 Motivation

In order for computer vision algorithms to be useful in the real world, they must be implemented in computer vision systems with real-time processing capability. A typical computer vision system consists of the following components: Video acquisition, Pre-processing, Feature Extraction, Processing, Decision Making and Displaying Result. In general, a computer vision system should use simple, reasonable, but reliable and fast computer vision algorithms. In addition, the system is usually implemented in the C/C++programming language and with hardware acceleration techniques, such as GPU-accelerated computing and multi-threading.

COCOALIGHT [6, 15] is a computer vision system for analyzing video data generated by UAVs. It can perform motion compensation, moving object detection, object tracking, and event indexing. The COCOALIGHT system is designed and implemented to facilitate tracking with near real-time latency for a VGA video. The limit of this system is that it is only suitable for aerial image processing.

KNIGHT [75] is another computer vision system for moving object tracking from a stationary video camera. In this system, background modeling is employed to quickly detect moving objects. The objects are modeled by a combination of color, shape and motion information. KNIGHT is capable of seamlessly tracking objects across multiple cameras. It is implemented in C++ and it is capable of running at 15 Hz for a VGA video. However, this system is designed to work only in low-resolution video and simple scenes.

In this chapter we will describe NONA, which we developed for analyzing a high definition, 360 degree view sensor developed by MIT-Lincoln Lab which was installed at Boston Logan Airport. Figure 6.1 shows how the sensor is mounted in an airport terminal. The sensor is designed for use in any environment where surveillance of large open areas is required such as public places, airport terminals, plazas, stadiums, parking lots, highways and so on.

The sensor has several key capabilities: a) 600 mega-pixel field of view with a frame rate of 5 fps. b) Digital pan, tilt, and zoom to capture any details in full 360 degree field of view. c) Photo-quality images with no motion blurs. Overall it provides much better video quality thus facilitating computer vision tasks.

We collected a high-resolution surveillance dataset (Airport dataset) from the raw video data captured by the ISIS cameras at Boston Logan Airport. This dataset contains 13 video sequences at different locations in the airport terminal. Each video sequence has 1000 frames, each of size 4000 by 2672. The frame rate of the videos is around 5 frame per second. We show some screen shots of this dataset in Figure 6.2.

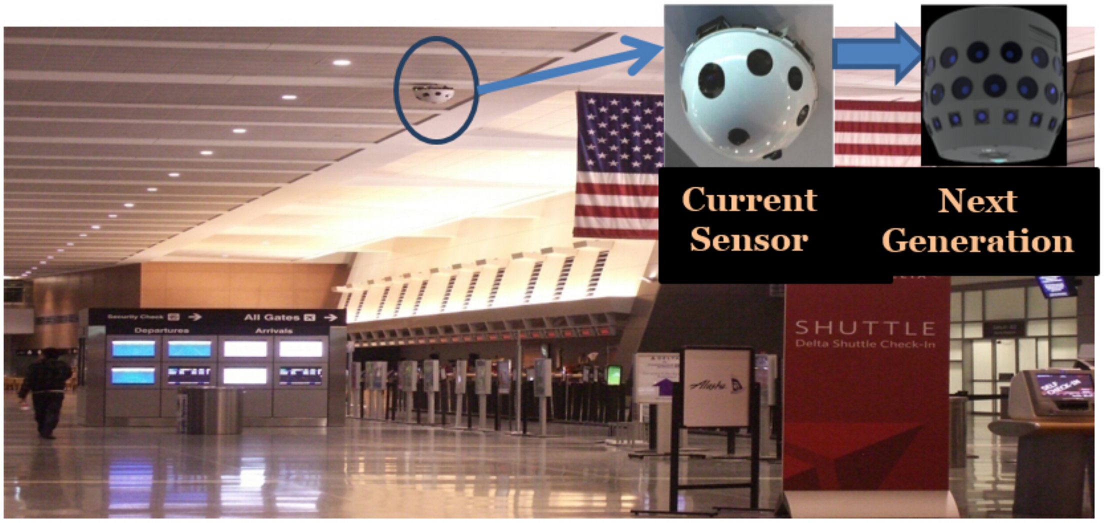


Figure 6.1: High-resolution surveillance sensor in Boston Logan Airport terminal. It has 48 cameras and full 360 degree field of view

Human tracking in this dataset is a very challenging task for a few reasons.

- The scene is semi-crowded, with partial or full occlusions frequently presented.
- The backgrounds are cluttered with significant illumination changes and shadows.
- The scale of the pedestrians is constantly changing as they walk towards or away from the cameras.
- The viewpoint of cameras varies from 30 degrees to directly below, which makes it difficult to apply a general human detector.
- The frame rate is relatively low.
- Most tracking methods cannot achieve real-time performance.

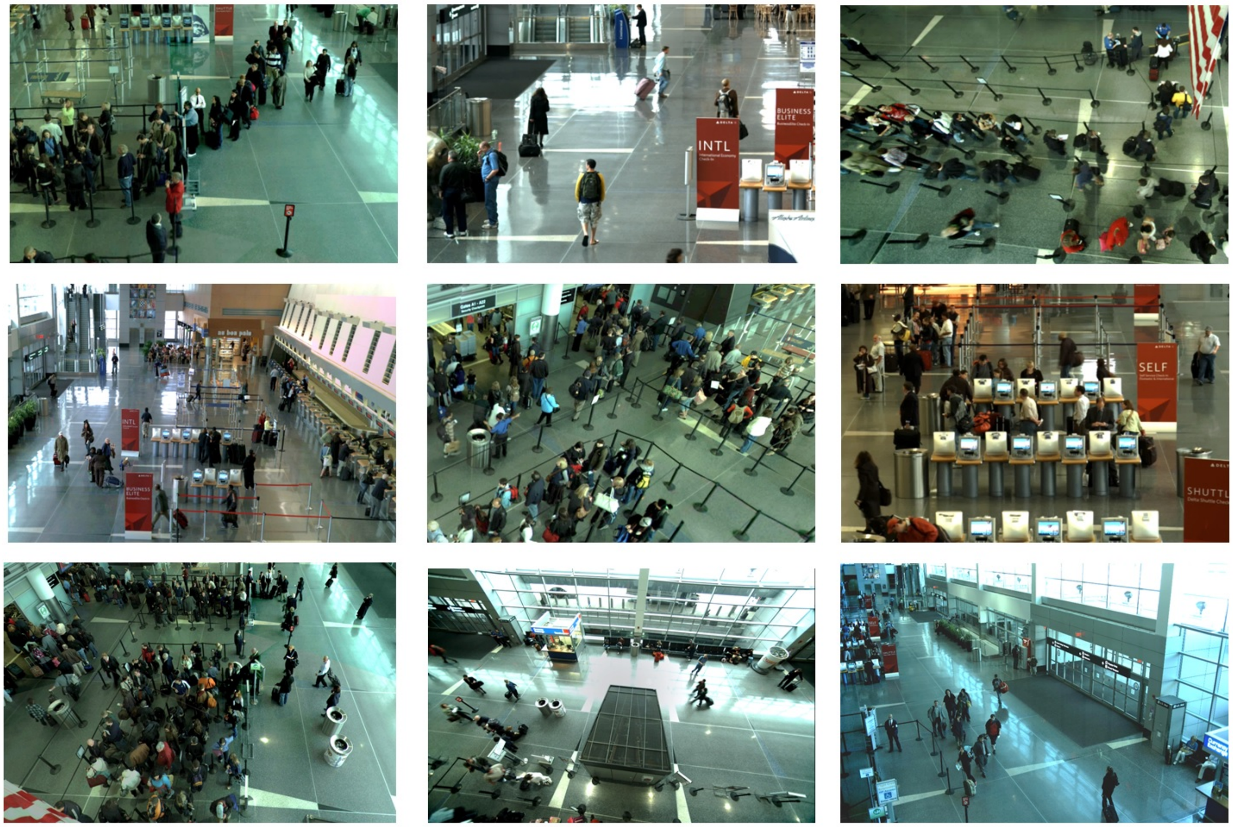


Figure 6.2: Sample images from several video sequences in the Airport dataset

We developed the NONA tracking system especially for this dataset. The key functionality in NONA is to be able to track a selected target in a crowd in high-definition video. Compared COCOALIGHT, NONA works with stationary cameras therefore motion compensation and sequence alignment steps are not needed. Also in NONA details of the object being tracked are more visible as the resolution is higher and so the simple correlation based tracker works well. The key challenge is computational cost, as the data volume is large with a typical frame size of 4000 by 2672, 5 frames per second. Other challenges include accuracy in the context of dealing with appearance similarity of nearby objects, occlusion etc.

6.1.1 System Overview

The NONA system is implemented with a multi-threaded architecture. The system essentially performs three main tasks: loading a frame from a video, tracking the target, and drawing the frame on the screen. We observed that the frame loading and drawing steps take over half of available CPU resources. Since these two steps are independent of the tracking process, we can leverage multi-threading to execute them in parallel with the tracking, thereby significantly enhancing performance.

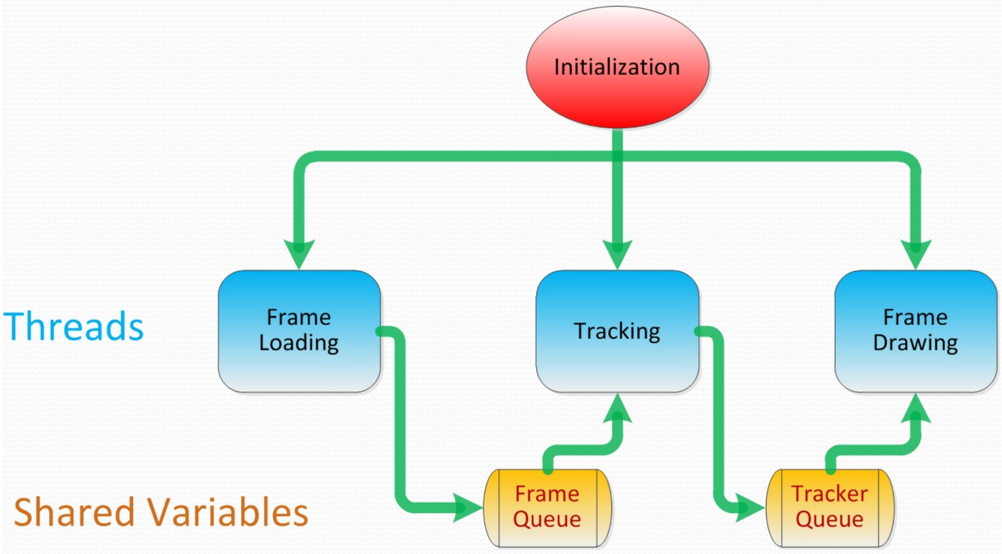


Figure 6.3: The multi-threaded NONA implementation. The three stages are executed separately as three distinct threads. Two queues of shared variables are used for thread communication.

The multi-threading framework is shown in Figure 6.3. We built the multi-threaded system using Intel Threading Building Blocks (TBB) [2]. Intel TBB is a library that helps developers easily build multi-threaded applications, taking full advantage of the multi-core processor performance.

The initialization includes loading the video and labeling the initial template. After the initialization, for each frame to be processed, the three stages are executed separately as three distinct threads. We use two queues of shared variables for thread communication. The Frame Loading thread will always load a frame to the Frame Queue unless it is full. The Tracking thread will fetch a frame whenever it is free and output the tracker position to the Tracker Queue. Finally the Frame Drawing thread will draw the frame and the tracker's location on the screen.

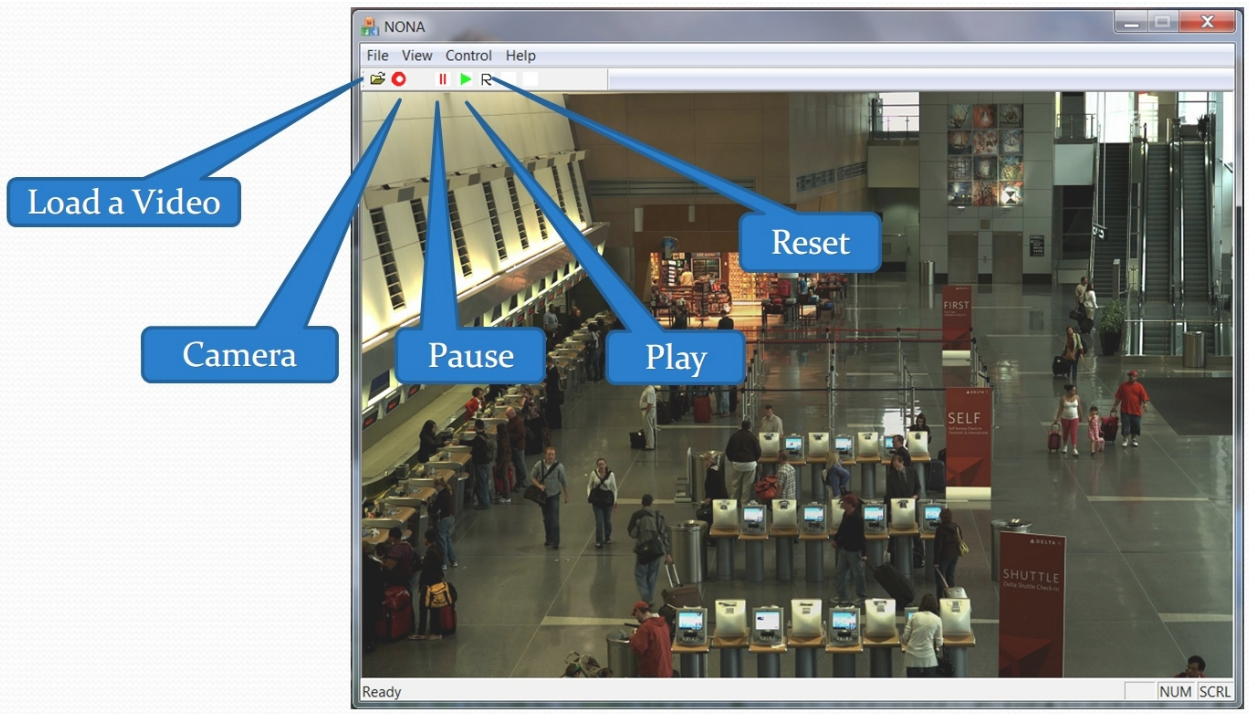


Figure 6.4: NONA system interface. It has several functions include loading from a video, online streaming from a camera, pause a video, play a video, and reset.

The interface is shown in Figure 6.4. NONA is designed for two working modes. In the streaming mode, NONA has the ability to perform real-time tracking though a streaming camera with high-resolution. In the offline mode, it can load archived videos with even higher resolution (up to 5000×4000) and work in a non-synchronous manner. User interactions is required for initializations. Once the video is displayed, the user manually labels a target by dragging a bounding box onto it. The user can switch to another target at any time. The tracker is always shown as a red bounding box in the screen.

NONA is developed with Visual C++, OpenCV [1] and Intel TBB. It runs on the following hardware configuration: 3 GHz CPU, 2 GB RAM and 4 cores. With the multi-threaded implementation, it is able to track a single object in high-resolution video (4000×2672) with a frame rate up to 9 fps, which is real-time for our tracking environment.

6.2 Proposed Tracking Algorithm

Our tracking algorithm can be formulated as a Bayesian inference framework. The target state can be obtained using the maximum a posteriori (MAP) estimation by Equation 6.1:

$$\hat{x_t} = \arg\max_{x_t} p(x_t \mid z_{1:t}) \tag{6.1}$$

where x_t is the target status at time t, and $z_{1:t} = \{z_1, z_2, ...z_t\}$ is all the observations up to time t. Using Bayesian theorem, the posterior probability $p(x_t \mid z_{1:t})$ can be recursively inferred by Equation 6.2 and 6.3:

$$p(x_t \mid z_{1:t}) \propto p(z_t \mid x_t) p(x_t \mid z_{1:t-1}) \tag{6.2}$$

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} \tag{6.3}$$

Based on the Bayesian inference framework, we proposed an efficient single-target tracking algorithm. The flowchart of our algorithm is shown in Figure 6.5.

Given the first image in the video sequence, we manually label the bounding box which contains the target. For each video frame, a searching region is predicted based on tracker's location with a learned motion model. Then a motion map in this region is detected by using the frame difference to exclude the background. Multiple features including intensity, hue, saturation and gradient are extracted from this region and masked by the motion map. The same features are also extracted from different scales of the template, which is initially manually cropped in the beginning of the first frame, and updated every frame. Then the target is located and the best template scale is selected by correlation-based template matching. Two important functions of NONA are the local frame differencing and adaptive template scaling shown in figure 6.5. The first prevents the tracker from drifting to background; the latter handles the size changing the issue of scale when the target moves towards the camera.

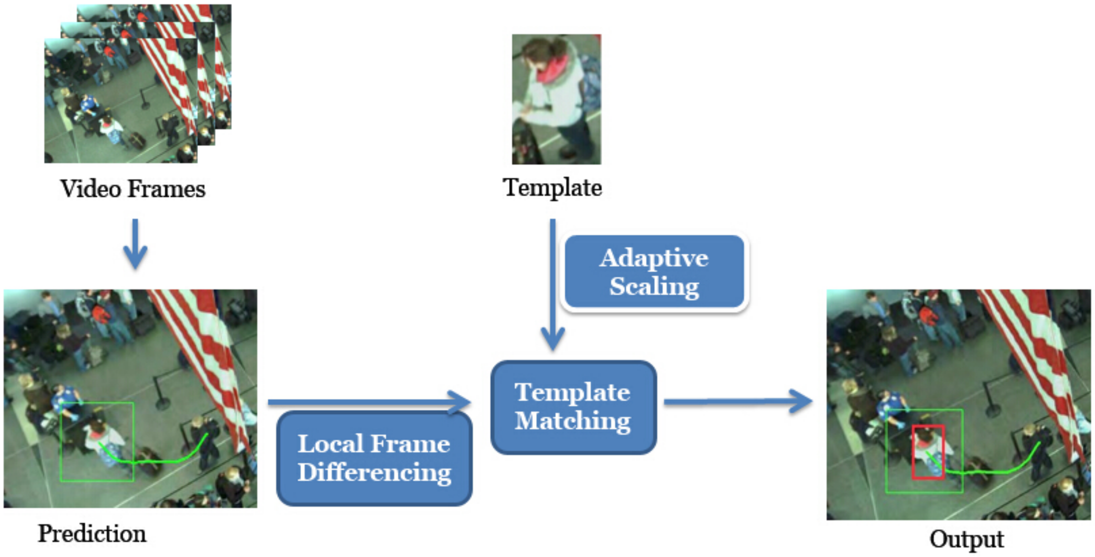


Figure 6.5: The flowchart of the proposed tracking algorithm

6.2.1 *Appearance Features*

Multiple features including Intensity, Hue, Saturation and Gradient are extracted from the initial and following bounding boxes. Figure 6.6 shows the features extracted from a pedestrian. It is worth noting that we chose these features mostly for computational efficiency. More sophisticated features such as HOG, SIFT or superpixels can also be used in our framework.

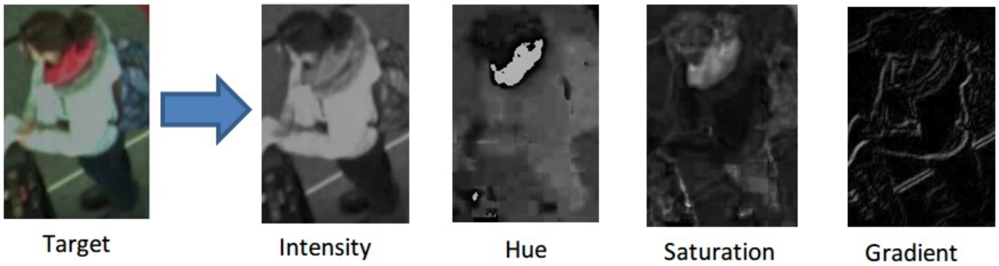


Figure 6.6: Four different features extracted from a target.

6.2.2 Motion Model

A linear motion model is used to predict the target's location p_t based on previous observed location and velocity:

$$p'_t = p_{t-1} + v_{t-1}, \tag{6.4}$$

where p'_t is the predicted target's location at time t, p_{t-1} is the observed location at time $t-1$, and v_{t-1} is the velocity at time $t-1$.

We predict a larger searching region which may contain the target at time t. Figure 6.7(b) shows the searching region in green bounding box. In our experiments, the searching region is set to 350×350, while a typical pedestrian bounding box is 100×200. Searching over a limited area is much faster than global searching, and it is more reliable to find the target in a smaller area than in the whole image. Once we have obtain the observed location p_t at time t, we can use it to update the motion model by Equation 6.5.

$$v_t = \alpha(p_t - p_{t-1}) + (1-\alpha)v_{t-1}, \tag{6.5}$$

where the velocity v_t is a weighed combination of current velocity and historical information which can smooth the motion model. α is the weight that controls the trade-off between them. We set

$\alpha = 0.7$ for most of our experiments. The initial velocity v_0 in the beginning is set to 0.

6.2.3 Template Matching

Given the searching region and the template, we calculate Normalized cross-correlation [63] of the template $t(x, y)$ with a sub-image f(x,y) in the searching region by Equation 6.6.

$$Ncc = \frac{1}{n} \frac{\sum_{x,y}(f(x,y) - \bar{f})(t(x,y) - \bar{t})}{\sqrt{\sum_{x,y}(f(x,y) - \bar{f})^2 \sum_{x,y}(t(x,y) - \bar{t})^2}} \tag{6.6}$$

where n is the number of pixels in $t(x, y)$, $\bar{f}, \bar{t}$ are the average of f, t respectively. We generate a response map M_i for each features i. Then all response maps are combined into one correlation score map M_c by Equation 6.7

$$M_c = \sum_{i=1}^{n} \omega_i M_i, \tag{6.7}$$

where ω_i is the weight of feature i. The combined correlation map is shown in Figure 6.7(c), where the maximum in this map (red star) reflects the location of most probably observation. In addition, we use a 2D Gaussian prior G centered at the predicted location to correct the Normalized correlation. Then we obtain the final score map which is the product of the correlation map and the Gaussian prior by Equation 6.8

$$M = M_i G, \tag{6.8}$$

A threshold Th is set to decide if the observation is reliable. If the maximum score of the observation is larger than Th, it is considered reliable, otherwise the predicted location is used instead. If a reliable matching score is found, the status of target is updated. Then a new template T_t is generated by combining both the new observation O_t and the previous template T_{t-1} by Equation 6.9.

$$T_t = \beta O_t + (1 - \beta) T_{t-1}, \tag{6.9}$$

where β is the updating weight. A large β can immediately capture the appearance change but it is sensitive to noises. In our experiments we set $\beta = 0.3$ for a more conservative updating strategy.

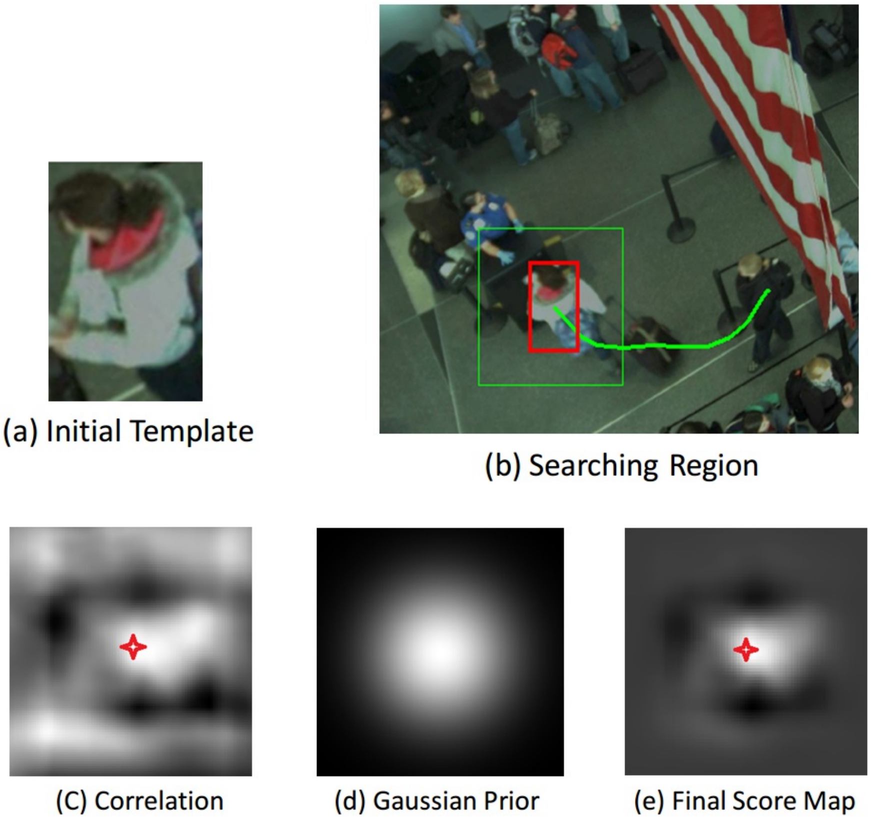


Figure 6.7: Template matching. (a)Initial template labeled in the first frame. (b)Matching results in frame 60: the green bounding box is the current searching region; the red bounding box is the match; the target track is shown in green. (c) Correlation surface using multiple features; the red star shows the peak of this surface. (d)Gaussian prior in the search region. (e) Final score map, which is the product of correlation surface and Gaussian prior; the red star shows the peak of this surface.

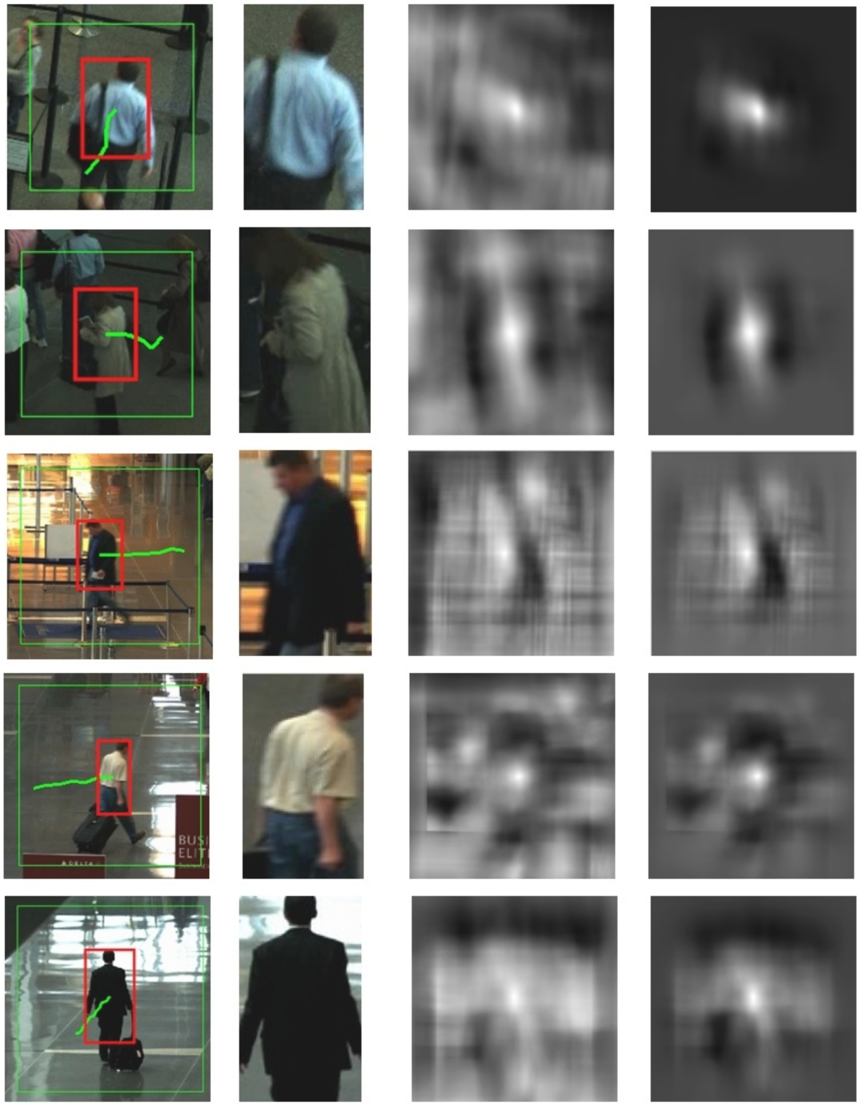


Figure 6.8: More template matching results. Each row shows the template matching for one person. The first column shows the original image. The second column shows the template, the third column shows the correlation surface. The fourth column shows the product of correlation surface and Gaussian prior

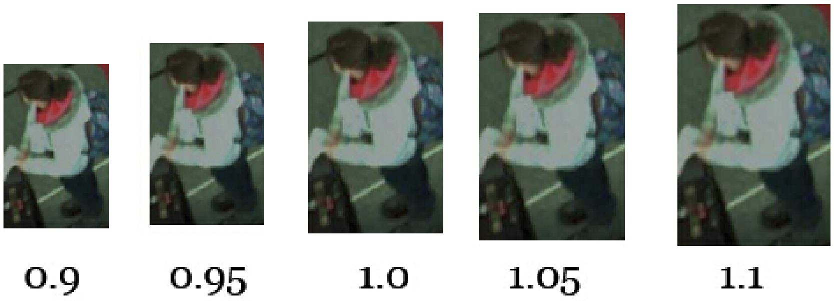


Figure 6.9: Various scales of a template created in each frame.

6.2.4 Adaptive Template Scaling

Since the cameras are set with various angles from 30 degree to directly below, the size of the target varies at different distances due to the perspective of the scene, which causes unstable matching. To resolve this problem, we use an adaptive template scaling method: Different scales of template S_t are created each time t. In our experiments we use 5 scales for efficiency as shown in Figure 6.9. We match all the scales on the searching region and preserve the optimal scale $\hat{S}_t$ with the maximum matching score by Equation 6.10.

$$\hat{S}_t = \arg\max_{S_t} M(S_t), \tag{6.10}$$

where $M(S_t)$ is the matching score of template S_t. The adaptive template scaling method can automatically track the size change of a moving target. Figure 6.10 shows a pedestrian with scale change is effectively tracked.

6.2.5 Occlusion Handling

In the Airport Terminal Dataset, one major difficulty is that the passengers are frequently occluded by static obstacles or other people. We explore two methods to resolve this problem.

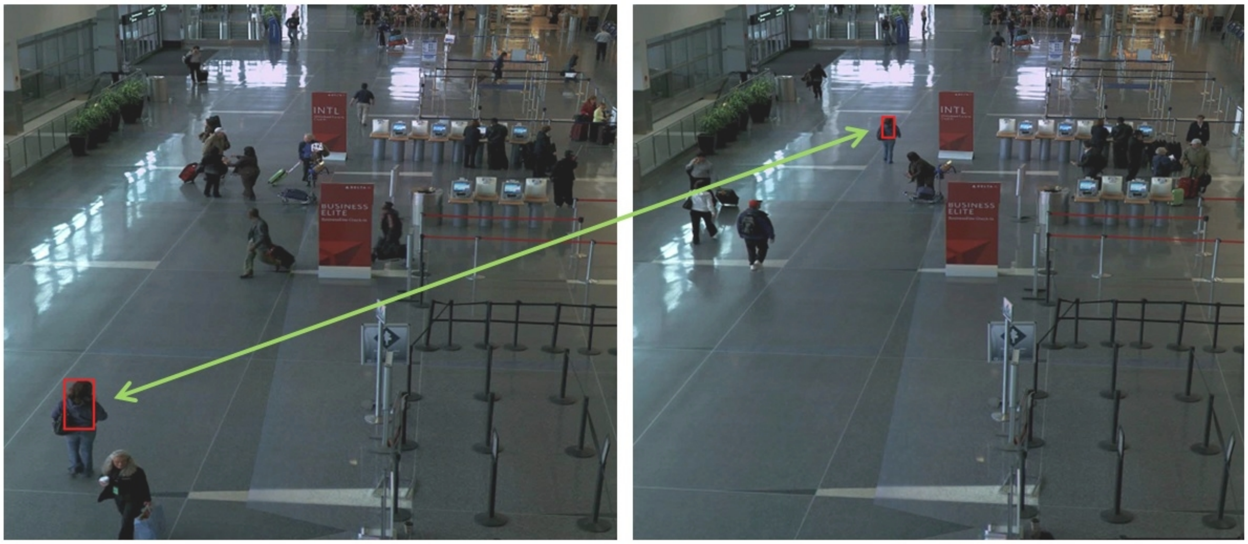


Figure 6.10: Adaptive Template Scaling can be used efficiently to deal with scale change. The red bounding boxes in the two images represent the same target at different time with significantly size changes.

6.2.5.1 *Incremental Searching*

In the template matching, if the matching score is lower than Th, we ignore the matching result and assume the target is lost due to occlusion. We will then leverage the motion model to predict the target's locations until it is observed again. In this case, we also gradually increase the size of the searching region considering the target may turn in a different direction. The maximum size of the searching region is set to 1/4 of the image in our experiment.

6.2.5.2 *Local Frame Differencing*

In many cases, once the target is lost, the tracker will drift to the background or other objects with similar appearance. To address this issue, we propose a local frame differencing technique to separate the foreground and background. Local frame differencing is to calculate difference between two consecutive frames in a small local area. From the local frame difference

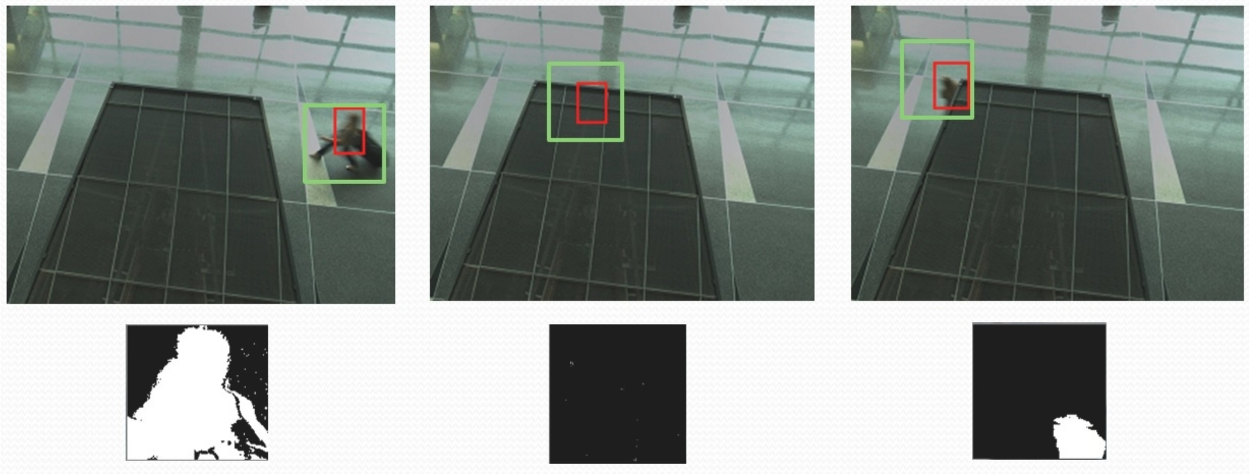


Figure 6.11: Local frame differencing techniques help to track the pedestrian with temporary occlusion. The first row shows the three video frames: before occlusion, in occlusion and after occlusion. The second row shows the image difference over the current searching regions. Note that when the target is occluded, no foreground area is detected so the tracker looks to the motion model for its estimate of trajectory.

we can obtain the foreground area which has non-zero values. This foreground area is then used as a mask for the template matching to prune the background area. The local frame difference is fast to calculate, but very effective in keeping the tracker away from the background.

In case the target is stationary, we execute a movement check: if the motion model shows a target is not moving, the local frame differencing step will be skipped. Combined with the Incremental Searching, this method can solve the persistent occlusion problems as shown in Figure 6.11.

Beside occlusion, local frame differencing can also improve tracking accuracy in some difficult cases, e.g. appearance change, sharp U-turn. More results are shown in Figure 6.12.

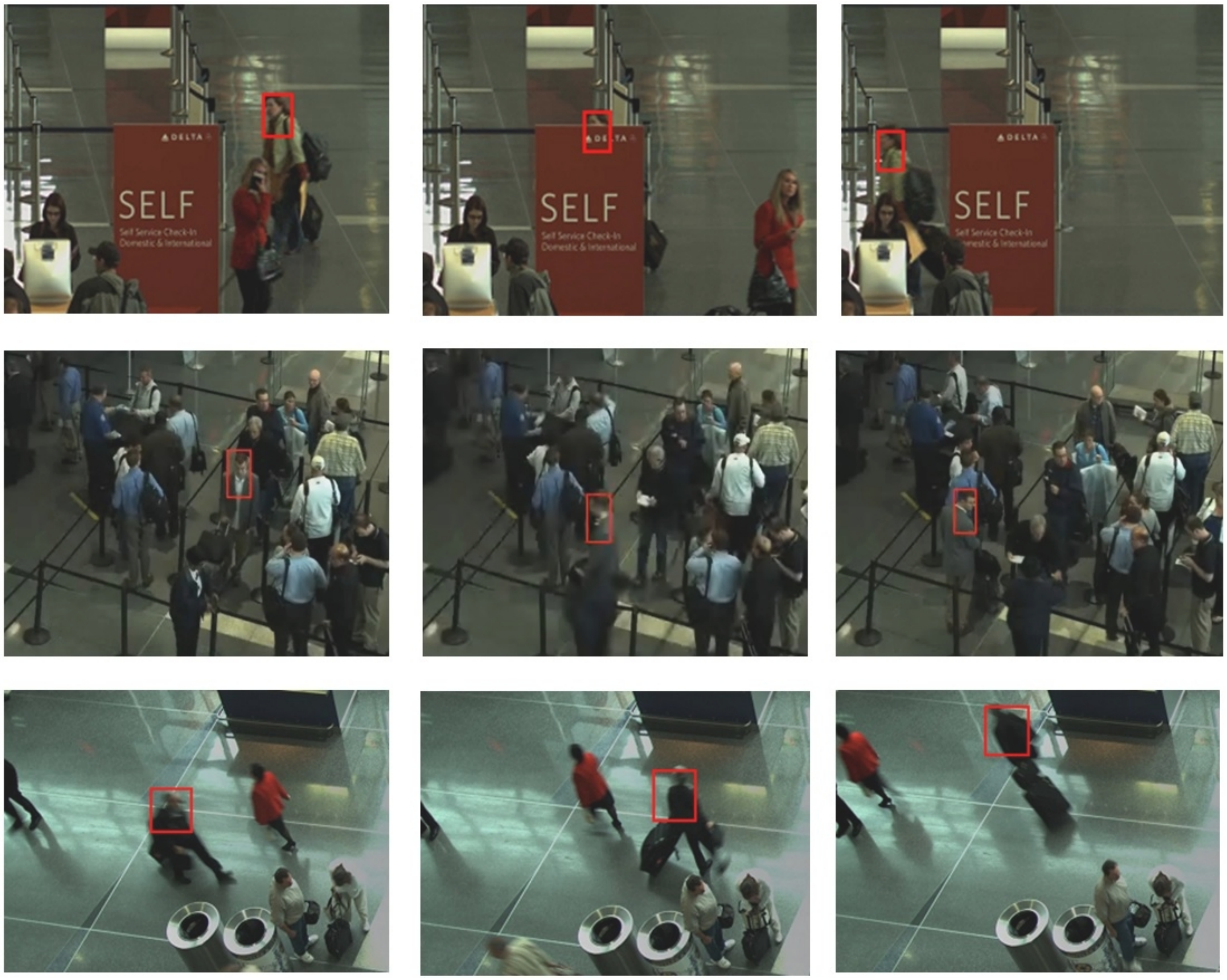


Figure 6.12: Tracking results of hard examples. The local frame differencing method helps in successfully tracking these targets.

6.3 Experimental Results

NONA has been tested on four indoor sequences of the Airport Dataset that are taken from different angles with roof-mounted cameras. In the experiment, we randomly label 100 testing samples from each sequence and track objects using NONA. The quantitative results are given in Table 6.1. The

Table 6.1: Tracking results of the NONA system on four Airport sequences. We use the following criterion to evaluate NONAs tracking performance for practical applications: The Accuracy in the table is the percentage of targets being successfully tracked over 70% of the actual tracks. Successful tracking is defined as the tracker having a 50% or more overlap with the object.

Dataset	Resolution	Accuracy	Down-sampling resolution	Accuracy
Airport-A	4096×2560	87%	640×400	76%
Airport-B	4096×2560	83%	640×400	70%
Airport-C	2048×2048	84%	512×512	75%
Airport-D	2048×2048	79%	512×512	68%

quantitative results are given in Table 6.1. Correct tracks represent the number of targets that have been successfully tracked over 50% of frames in a clip. Coverage is the average percentage of correctly tracked frames in a clip. In table 6.1, we made a quantitative comparison between the original video and down-sampling video. We observed original video significantly outperforms down-sampling video by 12.3% on average. This demonstrates that high-resolution video provide more discriminative and detailed features for the correlation-based tracker.

Figure 6.13 compares the correlation surfaces generated by high-resolution image and low resolution image. We observe that for the high-resolution image, the peak of the correlation surface is more distinctive, which improves the template matching accuracy.

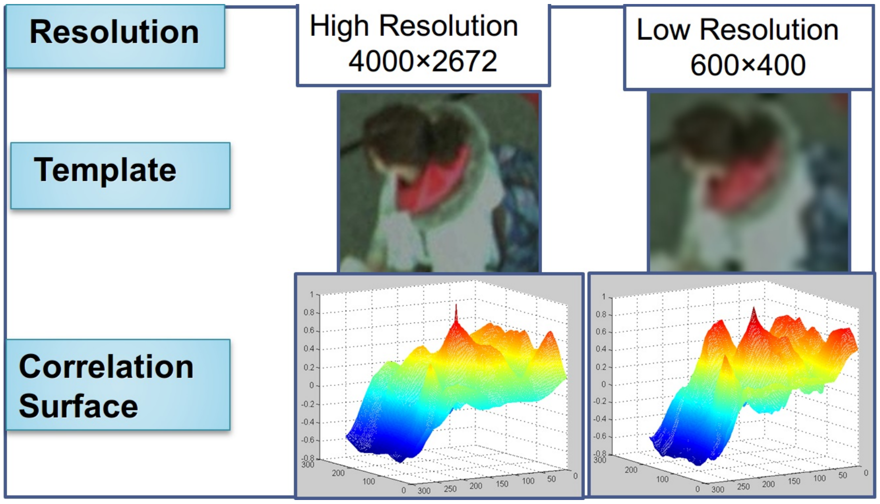


Figure 6.13: This figure compares the shape of correlation surfaces generated by high-resolution image and low resolution image. We observe that for the high-resolution image, the peak of the correlation surface is more distinctive, which improves the template matching accuracy.

We show some qualitative results in Figure 6.15, 6.16, 6.17 and 6.18. It can be seen that most of the targets are successfully tracked through hundreds of frames.

We also tested the computational efficiency of NONA using an Intel Quad Core 2.8GHz PC. The quantitative results are shown in table 6.2. We tested sequence A with different resolutions by sub-sampling at: the full resolution, quarter, and one-eighth the original size. We selected the same 10 targets in each different resolution, tracked them for 100 frames and computed the average time taken by each processing stage.

In table 6.2, columns 2-4 show the execute time for frame loading, drawing and tracking respectively. We observed frame loading and drawing take over half of the total execute time, therefore it is necessary to use multi-threading. Columns 5 and 6 show the comparison between single-threading and multi-threading. It is worth noting even in the highest resolution (4000×2672), the multi-threading implementation still has 9 fps, which achieves real-time performance.

Table 6.2: Computational efficiency of NONA system.

Frame Resolution	Frame Ingestion(ms)	Video output(ms)	Tracking (ms)	Singlethreading (frame/s)	Multithreading (frame/s)
4000 × 2672	72	81	62	5	9
2000 × 1336	21	30	39	11	21
1000 × 668	11	16	21	21	48

The NONA system is not limited to only tracking humans in the Airport Dataset, but is also capable of tracking any object in unconstrained videos. We have successfully tracked various objects in many videos with different resolutions, frame rates and image quality. It demonstrates that our NONA tracking system is a general tracking system which can track not only pedestrians, but also vehicles, baggage or animals in all kinds of video sequences. Figure 6.19 show some qualitative examples in other popular datasets. Figure 6.14 shows the NONA is tracking a cup using a streaming camera.

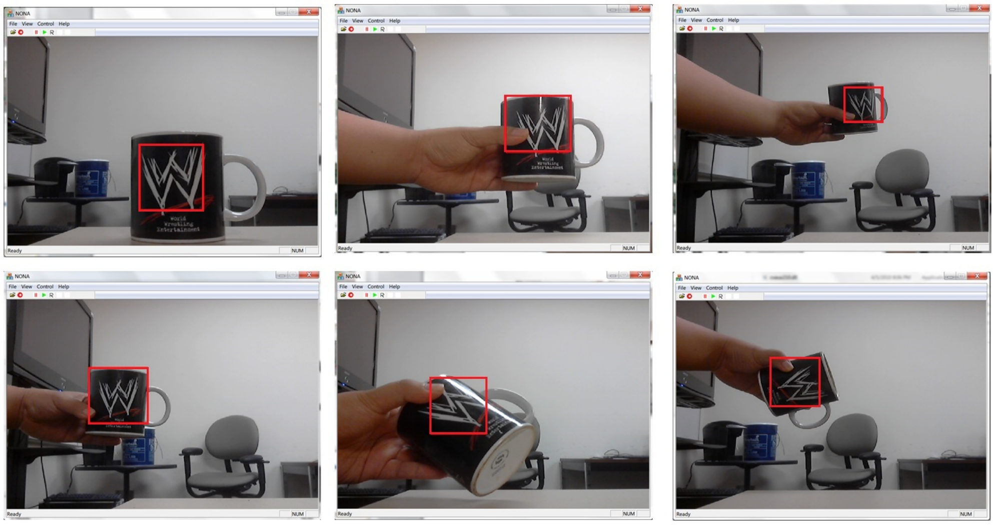
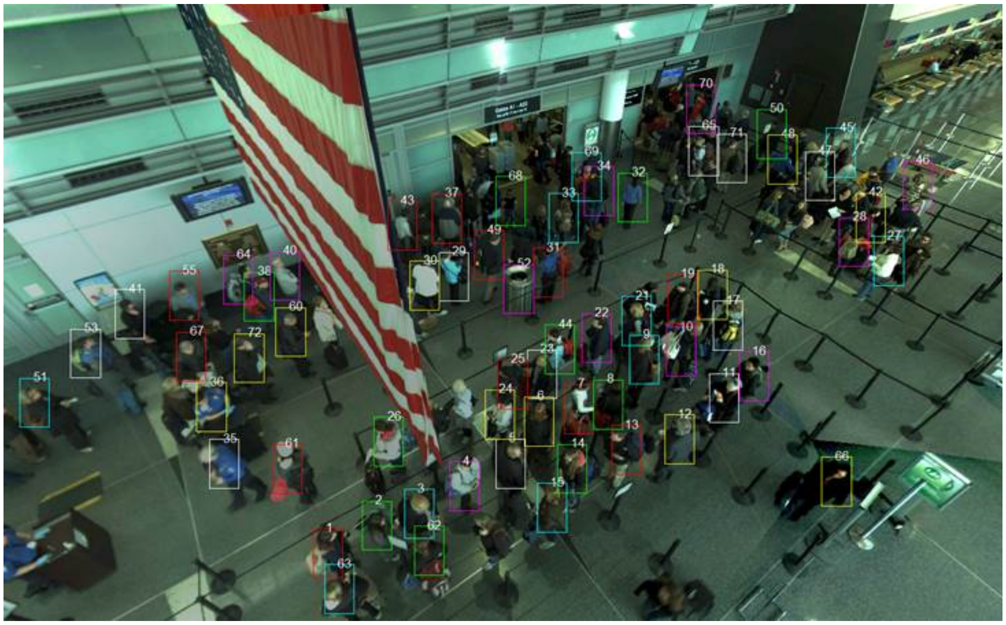
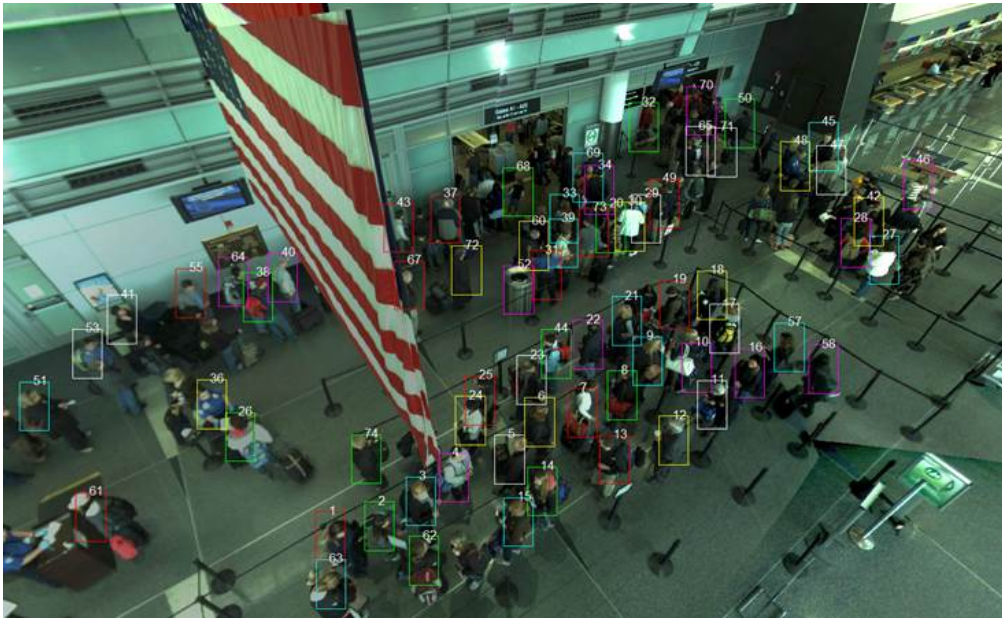


Figure 6.14: Tracking a cup using a streaming video.

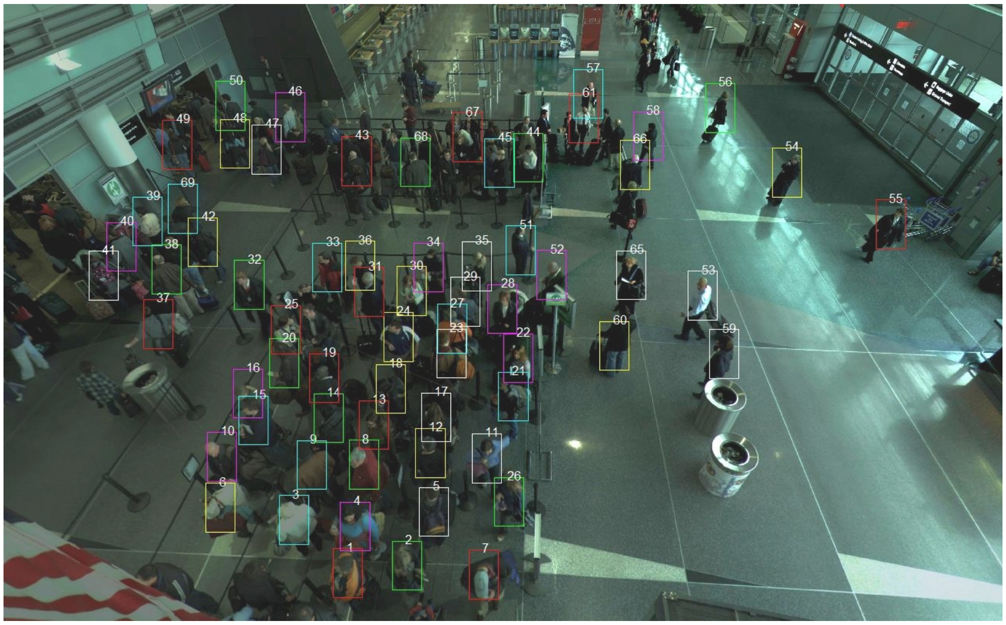


Frame 1 of Airport-A Sequence.

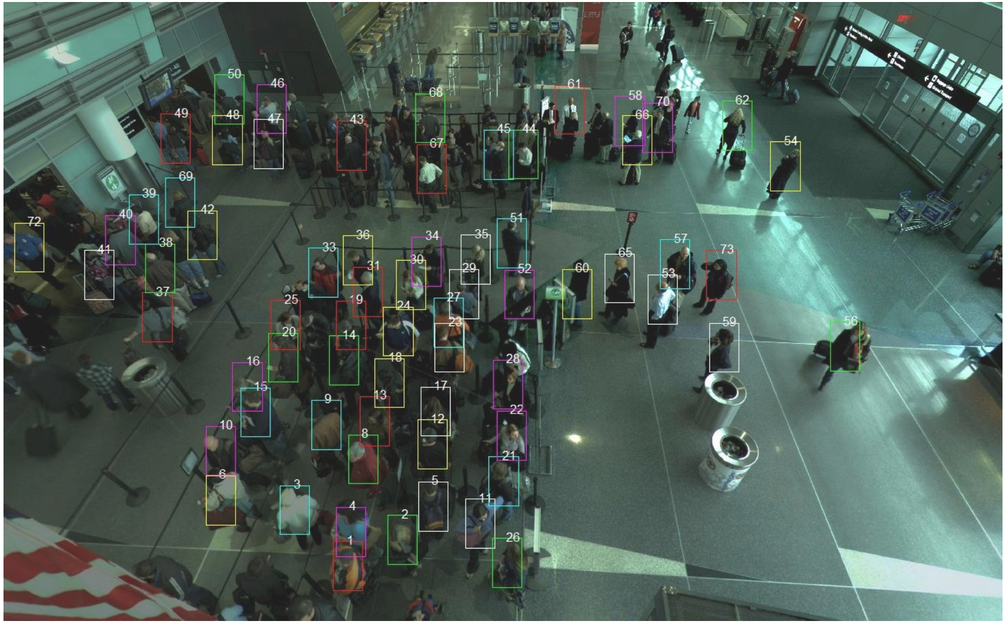


Frame 63 of Airport-A Sequence.

Figure 6.15: Tracking results on Airport-A sequence. We labeled and tracked all the targets and show all the bounding boxes. The number of the bounding box identify unique tracks across different frames.



Frame 10 of Airport-B sequence.



Frame 50 of Airport-B sequence.

Figure 6.16: Tracking results on AirportB sequence. We labeled and tracked all the targets and show all the bounding boxes. The number of the bounding box identify unique tracks across different frames.

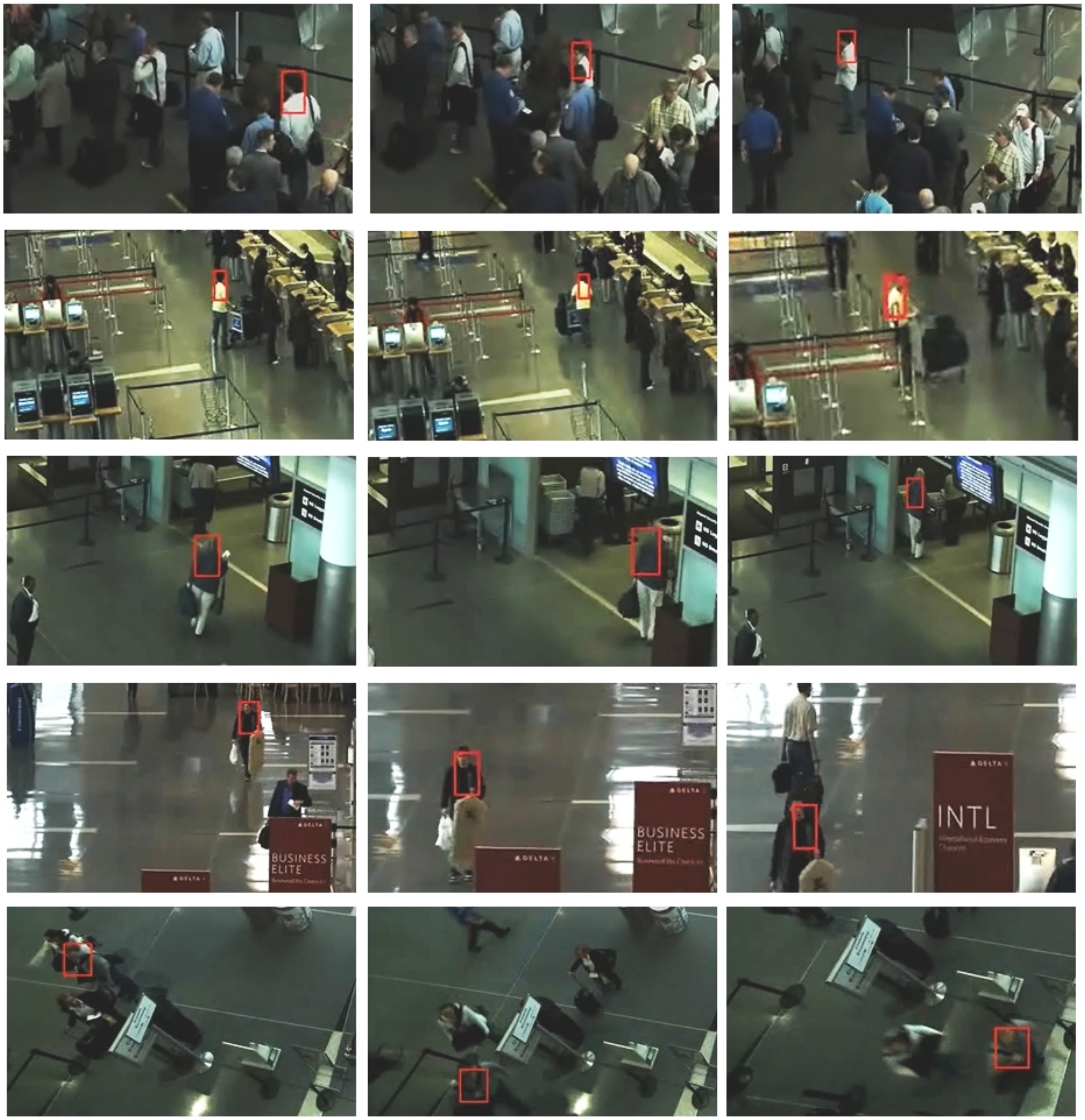


Figure 6.17: Tracking results using NONA. Each row shows a preson is being tracked.

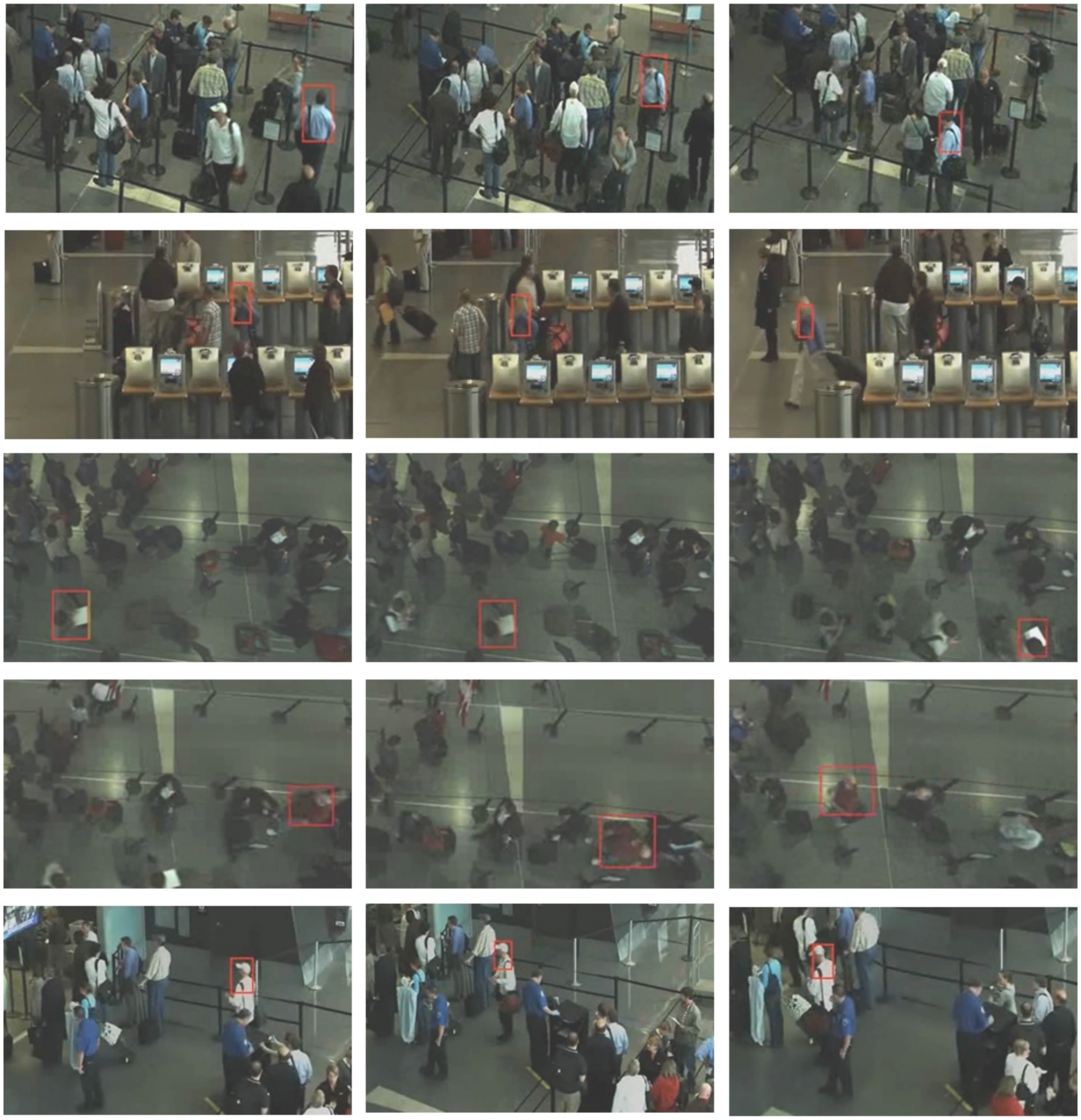


Figure 6.18: More tracking results using NONA. Each row shows a preson is being tracked.

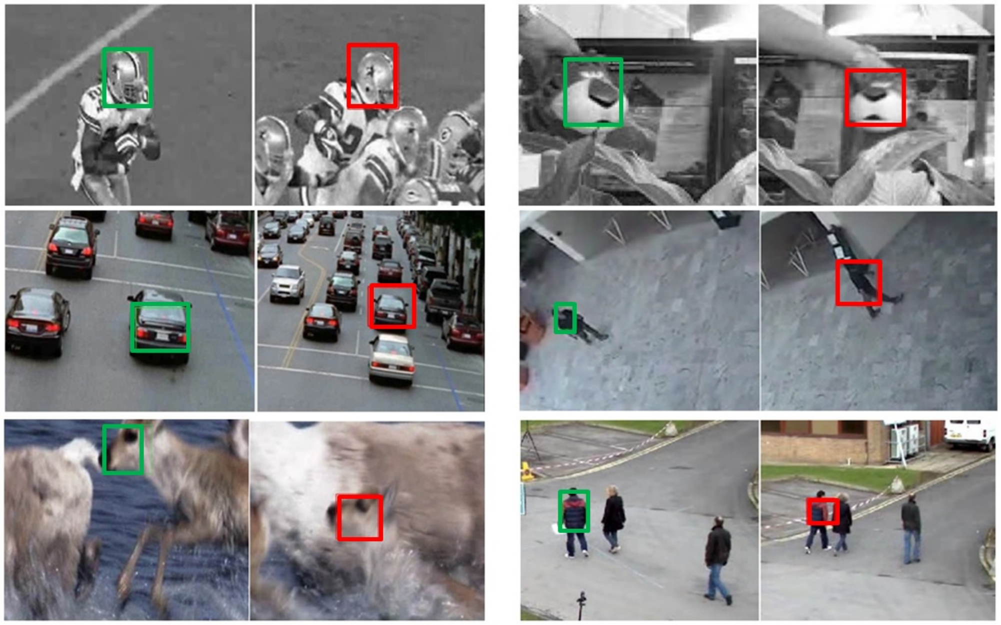


Figure 6.19: Tracking results of 6 popular datasets. The green bounding boxes show the initializations. The red bounding boxes show the tracker's location after some frames.

6.4 Summary

In this chapter we introduced an efficient tracking system, NONA, for high-resolution videos in semi-crowded environments. We employed efficient appearance features and FFT-based correlation algorithm to achieve real-time performance. In addition, we proposed several techniques including adaptive template scaling, incremental searching and local frame differencing to resolve challenging issues such as scale change and occlusions. In addition, we used a multi-threading implementation to further improve the tracking efficiency. Finally we extensively tested our tracking system on the Airport Dataset and achieved satisfying tracking accuracy while maintaining real-time performance.

CHAPTER 7: CONCLUSION AND FUTURE WORK

In this dissertation, we addressed the problems of human detection, tracking and segmentation in surveillance videos. These problems are challenging in that surveillance videos typically contain crowded environments, cluttered backgrounds, humans with different appearances, illumination changes and frequent occlusions.

The first problem we addressed was that of adapting a human detector in unconstrained videos. We used a semi-supervised learning framework to learn automatically scene-specific information in the video. The first step is to obtain training examples from the video using a pre-trained detector. Then a new classifier is trained with a superpixel-based Bag-of-Words representation that encodes the color information. This classifier is trained using new examples at each iteration, gradually improving the detection precision. The results of the experiment show that the proposed method outperforms the DPM detector in average precision by up to 16%.

The second problem we addressed was that of tracking multiple-humans in surveillance video captured by a single camera. We proposed an online-learning tracking-by-detection framework which learns a discriminative classifier for each individual pedestrian. In this tracking framework we firstly employed a DPM human detector to obtain detections, and then associated detections across frames by using the similarities in position, size and appearance. We also proposed additional occlusion handling methods in both the detection and tracking stages. In the detection stage we inferred the part scores of the DPM detector to find possible partial occlusions. In the tracking stage we inferred the part scores of the classifier dynamically to overcome the occlusion and predict future occlusions. We also put together a PNNL parking lot dataset for testing the tracking methods. The experiments showed satisfactory performance for our method in semi-crowded datasets.

The third problem we addressed was that of multiple human segmentation in videos. To solve this problem, we proposed a novel method based on detection. We estimated the initial segmentation based on superpixels and detection potential. The detection potential is obtained using the response of the part-based detector, which provides useful human shape estimation. After we obtained an initial segmentation of the object, we built a tracklet-based Gaussian Mixture Model and obtained smooth boundaries using multi-layer graph optimization. In a given video sequence, our method can segment all the humans automatically; it does not require manual interaction and is not sensitive to camera motion. We annotated the segmentation ground truth for six datasets in order to test our method. The results of the experiment showed that our method significantly outperforms previous human segmentation methods.

The fourth problem we addressed was the challenge of building a real-time tracking system for high-resolution videos. For computation efficiency, we chose several very efficient algorithms in our tracking system including a Bayesian inference framework, a normalized cross-validation algorithm and a linear motion model. Multi-threading architecture was implemented in our system to improve computational efficiency significantly. In order to robustly track a target in a crowded environment, we used the Local Frame Differencing technique to handle long term occlusions, and the Adaptive Template Scaling technique to handle the scale change. We also put together an Airport video dataset using a state-of-the-art high-resolution surveillance sensor.

7.1 Summary of Contributions

Our main contributions are summarized below.

1. Improving a human detector using superpixels.
    (a) An algorithm to improve a generic human detector using an unsupervised learning framework.
    (b) Representation of humans in terms of superpixel-based Bag-of-Words representation.
2. Part-based multiple-human tracking with occlusion handling

(a) An algorithm to track multiple humans using online-learned person-specific classifiers.

(b) Representation of humans in terms of part-based model to capture the appearance variations.

(c) An occlusion handling approach to improve human detection performance in crowded scenes.

(d) An approach to overcome partial occlusions for human tracking.

3. Multiple-human segmentation leveraging human detector

   (a) An algorithm to automatically segment multiple humans in videos based on human detections.

   (b) Representation of humans in terms of the part-based detection potentials to capture the spatial distribution

   (c) A new approach to using tracklet-based CRF optimization to smooth the segmentation boundaries.

4. NONA: An efficient tracking system

   (a) A computer vision system for real-time human tracking in high-resolution videos.

   (b) A multi-threaded architecture to process video ingestion, tracking and video output simultaneously.

   (c) A novel approach to using local frame differencing to handle long-term occlusion.

7.2 Future Work

This section explores some of the possible directions our work may be taken in the future.

Motion-based human detection Our proposed human detection method used only static features such as superpixels or HOG, while in videos the motion features, e.g. Histogram of

Optical Flow (HOF), can also provide very discriminative information. Since the local motion patterns of humans are very different from the background or other objects, these motion features are very helpful for human detection in videos. To apply them successfully we will need to take into account issues such as the quality of optical flow and camera motion.

Tracking humans across multiple cameras Another direction for human tracking is leveraging multiple cameras. The proposed human tracking algorithm is based on a single camera. However, multiple cameras are widely used in video surveillance. These cameras can cover the entire surveillance region, with no overlaps or very small overlaps among them. Tracking a target across different cameras would be a very useful application for wide area surveillance. The challenge is to correctly associate a person appearing in different cameras, each with differing viewpoints and lighting, whilst considering the spatial and temporal constraints.

Human pose estimation based on segmentation One interesting area to explore is estimating human poses in videos. Human pose estimation can benefit other computer vision tasks such as human-computer interaction and activity recognition. The proposed human segmentation algorithm provides accurate and smooth human silhouettes, which can be used to estimate the poses. Human pose estimation can be formulated as a multi-class classification problem: given the segmentation as an input, we can extract features, train a classifier and predict the class label. The essential step is to design a good feature that is extracted from the boundary and that can distinguish different poses.

LIST OF REFERENCES

[1] http://opencv.org/.

[2] https://www.threadingbuildingblocks.org/.

[3] Pets 2009 benchmark data, http://pets2009.net/.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels. In *EPFL Technical Report*, 2010.

[5] T. Ahonen, A. Hadid, and M. Pietikinen. Face description with local binary patterns: Application to face recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, pages 2037–2041, 2006.

[6] S. Ali and M. Shah. Cocoa - tracking in aerial imagery. In *SPIE Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications*, 2006.

[7] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[8] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *CVPR*, 2013.

[9] S. Avidan. Ensemble tracking. In *PAMI*, 2010.

[10] R. N. B. Yang, C. Huang. Segmentation of objects in a detection window by nonparametric inhomogeneous crfs. In *Computer Vision and Image Understanding*, 2011.

[11] B. Babenko, M. Yang, and M. Hsuan. Visual tracking with online multiple instance learning. In *PAMI*, 2010.

[12] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Transactions on Graphics (TOG)*, 28(3):70, 2009.

[13] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011.

[14] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011.

[15] S. Bhattacharya, H. Idrees, I. Saleemi, S. Ali, and M. Shah. Moving object detection and tracking in forward lookinginfra-red aerial imagery. In *Mach. Vision Beyond Visible Spectr.*, 2011.

[16] S. Blackman. Multiple hypothesis tracking for multiple target tracking. In *IEEE Aerosp. Electron. Syst. Mag.*, 2004.

[17] A. Blake and M. Isard. Active contours: The application of techniques from graphics, vision. In *Control Theory and Statistics to Visual Tracking of Shapes in Motion*, 1998.

[18] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. In *International Journal of Computer Vision*, volume 70, pages 109–131, 2006.

[19] Y. Boykov and Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, 2001.

[20] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009.

[21] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011.

[22] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, pages 833–840. IEEE, 2009.

[23] C. R. Brice and C. L. Fennema. Scene analysis using regions. In *Artificial Intelligence*, volume 1, pages 205–226, 1970.

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

[25] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295. Springer, 2010.

[26] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010.

[27] H. Celik, A. Hanjalic, and E. A. Hendriks. Unsupervised and simultaneous training of multiple object detectors from unlabeled surveillance video. In *Computer Vision and Image Understanding*, 2009.

[28] R. Collins and Y. Liu. Online selection of discriminative tracking features. In *ICCV*, 2003.

[29] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 603–619, 2002.

[30] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. In *PAMI*, 2003.

[31] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, volume 20, page 273, 1995.

[32] Q. Dai and D. Hoiem. Learning to localize detected objects. In *CVPR*, 2012.

[33] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[34] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of ow and appearance. In *ECCV*, 2006.

[35] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR*, 2011.

[36] P. Dollar, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

[37] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34, 2012.

[38] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *Computer VisionECCV 2000*, pages 751–767, 2000.

[39] M. Everingham, V. Gool, L. Williams, C. K. I., J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. In *International Journal of Computer Vision*, 2010.

[40] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[42] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[43] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *PAMI*, 2010.

[44] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *PAMI*, 2010.

[45] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. In *IJCV*, 2004.

[46] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. In *IEEE Transactions on Computers*, volume 22, page 6792, 1973.

[47] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009.

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

[49] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, 2006.

[50] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2141–2148. IEEE, 2010.

[51] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[52] Y. Huang, Q. Liu, and D. Metaxas. Video object segmentation by hypergraph cut. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1738–1745. IEEE, 2009.

[53] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 22–27. IEEE, 2002.

[54] Z. Kalal. P-n learning?: Bootstrapping binary classifiers by structural constraints. In *ICCV*, 2010.

[55] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *International Journal of Computer Vision*, volume 1, pages 321–331, 1988.

[56] C. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010.

[57] C. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010.

[58] C. Kuo and R. Nevatia. Robust multi-view car detection using unsupervised sub-categorization. In *WACV*, 2009.

[59] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *ICCV Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*, 2011.

[60] Y. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002. IEEE, 2011.

[61] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *CVPR*, 2003.

[62] LevinshteinA., A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. In *PAMI*, 2009.

[63] J. P. Lewis. Fast normalized cross-correlation. In *Industrial Light and Magic*.

[64] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[65] K.-C. Li, H.-R. Su, and S.-H. Lai. Pedestrian image segmentation via shape-prior constrained random walks. In *Advances in Image and Video Technology*, pages 215–226. Springer, 2012.

[66] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 670–677. IEEE, 2012.

[67] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004.

[68] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 696–710, 1997.

[69] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23, pages 349–361, 2001.

[70] G. Mori. Guiding model search using segmentation. In *ICCV*, 2005.

[71] S. Munder, C. Schnorr, and D. M. Gavrila. Pedestrian detection and tracking using a mixture of view-based shape–texture models. *Intelligent Transportation Systems, IEEE Transactions on*, 9(2):333–343, 2008.

[72] V. Nair and J. Clark. An unsupervised, online learning framework for moving object detection. In *CVPR*, 2004.

[73] R. Ohlander, K. Price, and D. R. Reddy. Picture segmentation using a recursive region splitting method. In *Computer Graphics and Image Processing*, volume 8, pages 313–333, 1978.

[74] O.Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progresively improving detectors. In *CVPR*, 2005.

[75] K. S. Omar Javed and M. Shah. Automated surveillance in realistic scenarios. In *IEEE MultiMedia*, 2007.

[76] D. M. P. Felzenszwalb, R. Girshick. Cascade object detection with deformable part models. In *CVPR*, 2010.

[77] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *ICCV*, 1998.

[78] O. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. The truth about cats and dogs. In *ICCV*, 2011.

[79] O. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Cats and dogs. In *CVPR*, 2012.

[80] T. Pavlidis and Y.-T. Liow. Integrating region growing and edge detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 12, pages 225–233, 1990.

[81] S. Pellegrini, A. Ess, , and L. van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010.

[82] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[83] B. Price, B. Morse, and S. Cohen. Livecut: Learning based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009.

[84] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[85] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In *ICCV*, 2011.

[86] A. Rosenfeld and L. S. Davis. Image segmentation and image models. In *Proceedings of the IEEE*, volume 67, pages 764–772, 1979.

[87] P. Roth, S. Sternig, H. Grabner, and H. Bischof. Classifier grids for robust adaptive object detection. In *CVPR*, 2009.

[88] C. Rother, V. Kolmogorov, and A. Blake. Grabcut -interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.

[89] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[90] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 8, pages 888–905, 2000.

[91] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[92] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*, 2012.

[93] G. Shu, A. Dehghan, and M. Shah. Improving an object detector and extracting regions using superpixels. In *CVPR*, 2013.

[94] H. Sidenbladh and M. J. Black. Learning the statistics of people in images and video. In *International Journal of Computer Vision*, volume 54, pages 189–209, 2003.

[95] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[96] K. Toyama, B. Krumm, J.and Brumitt, and B. Meyers. Wallpower: Principles and practice of background maintenance. In *ICCV*, 1999.

[97] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *ECCV*, 2008.

[98] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003.

[99] P. A. Viola and M. J. Jones. Robust real-time face detection. In *International Journal of Computer Vision*, volume 57, pages 137–154, 2004.

[100] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010.

[101] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *ECCV*, 2004.

[102] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011.

[103] X. Wang and T. Han. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009.

[104] X. Wang, G. Hua, and T. X. Han. Detection by detections: Non-parametric detector adaptation for a video. In *CVPR*, 2012.

[105] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. In *International Journal of Computer Vision*, 2007.

[106] B. Wu, R. Nevatia, and Y. Li. Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In *CVPR*, 2008.

[107] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg. Who are you with and where are you going? In *CVPR*, 2011.

[108] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Journal of Computing Surveys*, 38(4), 2006.

[109] C. Zhang, R. Hamid, and Z. Zhang. Taylor expansion based classifier adaptation: Application to person detection. In *CVPR*, 2008.

[110] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.

[111] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.