DETECTING PEDESTRIANS USING MOTION PATTERNS: A LATENT TRACKING APPROACH

by

Amirhossein Bakhtiarikouhsorkhi

M.Sc, University of Tehran, 2011

B.Sc, University of Tehran, 2008

A Thesis submitted in partial fulfillment
of the requirements for the degree of
MASTER OF SCIENCE

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2013

APPROVAL

Name:	Amirhossein Bakhtiarikouhsorkhi
Degree:	MASTER OF SCIENCE
Title of Thesis:	DETECTING PEDESTRIANS USING MOTION PATTERNS: A LATENT TRACKING APPROACH
Examining Committee:	Dr. Ted Kirkpatrick Chair
	Dr. Greg Mori, Senior Supervisor
	Dr. Mark Drew, Supervisor
	Dr. Ghassan Hammraneh, SFU Examiner
Date Approved:	December 20, 2013

Partial Copyright Licence

Abstract

In this thesis we present a new method to detect pedestrian in video sequences. Unlike most of the common detection methods which only rely on the appearance of an object for detection, our proposed method uses the motion information of the object as well as its appearance to improve the detection quality. The idea is to capture the motion of each body part and use these motion patterns and add them to an appearance based detector to improve the detection results. We try to model the distinct motion patterns of pedestrians using a set of latent variables. The proposed method is tested on two separate datasets and the quantitative results are outperforming two of the commonly used pedestrian detectors in the literature and showing the capacity of motion patterns to improve the detections.

Keywords: Pedestrian detection, Motion patterns, Object detection, Computer vision

To my parents for their endless love and support.

Acknowledgments

I would like to take advantage of this opportunity to appreciate numerous people who influenced this thesis and my studies at Simon Fraser University. I am truly grateful to my supervisor, Dr. Greg for his patience, support, encouragement, and help in both my education and life. I have been fortunate to work in Vision and Media Laboratory where constructive and supportive students and faculty are gathered. Specially, I wish to express my gratitude to Arash Vahdat, Kevin Cannons and Hossein Hajimirsadeghi for their generous discussion and genuine directions.

Last but not least appreciations go to my parents for their supports and encouragements in every step of the way.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

As the computation power got cheaper and cheaper in the last decades and capturing images and videos became a simple push of buttons, computer vision became very important and everyday researchers try to introduce methods to automatically perceive the world through cameras. Computer vision widely discusses object recognition, digital photography, automatic surveillance, navigation, content based media retrieval and in general any application whose goal is to replace or improve human visual and recognition system with a (semi-) automatic system.

One of the first steps in many machine vision applications is to detect and track an object in the input video stream or static image, in this thesis we introduced a new method for detecting pedestrians in videos. Detection of humans using video stream data is usually very challenging due to various poses, different clothing and partial occlusion of the subjects. To accomplish this task many methods and features are introduced in the machine vision and machine learning community. Unfortunately, most of the current methods in the literature use information from static images and ignore the information about the motion of the subject which carries a lot of information about the moving object, we tried to capture and use this information to improve our detection results.

1.1.1 Human detection

Computer-vision based analysis of videos is a broad active area of research. To solve this problem many approaches are introduced in the literature, these solutions stretch from detecting very low level features (color, motion, texture, etc) and combining them to do the classification to high level approaches where they try to detect high level features (actions, concepts, etc) and use them to do the classification. It seems that the human mind does something in the middle where it tries to detect objects and then combines the object detection with other information such as location, motion, etc to do the classification, Serre et al. [22]. One of the most important objects of interest is humans, since we have a human in most of the activities and we are interested in a reliable human detector. The solution to this problem can be used in automatic surveillance, computer game industry and content based video retrieval. Surveillance is usually needed in secured areas such as airports to decrease the emergency service arrival time. In video retrieval we usually interested to retrieve videos where people did action of interest and to do that we first have to detect humans in the videos.

In this thesis we focused on detecting pedestrians in videos, especially surveillance videos where we usually have videos viewing pedestrians from steep downward angle where pedestrians can be covered by each other, vehicles, tress and all other structures in an urban environment, these occlusions plus different clothing styles and different items which a pedestrian carries in an urban environment makes human detection in surveillance videos very difficult. Another difficulty in detecting pedestrians in surveillance cameras is that usually the camera is far away from the subject so the size of the pedestrian might become very small(e.g. 20-30 pixels in height) which results in poor performance for most of the common human detectors.

An important observation is that although the shape of the humans is very informative and can help to detect pedestrians, its usefulness is drastically reduced when the subject is occluded or when it is far away from the camera. Another cue which seems to be very helpful in detecting humans is the motion patterns of the humans, these patterns are very distinctive and they are usually visible and distinct even if the subject is partially occluded. In this thesis we are introducing a new method for human detection using motion patterns. Since these motion patterns are hard to detect and formulate we treat them as latent variables in our model and use the latent Support Vector Machine (SVM) [15] to do the detection. The

results of this method shows improvements in the detection of pedestrians over the baselines especially when the subjects are partially occluded.

The approach we took in this thesis is to divide the bounding box of the pedestrian into different parts which would roughly cover different parts of body (hands, feet and head). Then we track each part in subsequent frames which will capture the motion patterns of different parts of the body. We do the joint tracking which means that the path of different parts are related to each other for instance the head can't go leftward while the body is going rightward.

The main contribution of this thesis is to propose a model to capture motion patterns and use these motion patterns to improve human detection. To model the motion patterns we track different parts of body and then use the obtained tracklet and the relation between these tracklets as an indicator of the motion patterns.

Chapter 2

Literature Review

2.1 Introduction

In this chapter we will review some of the most successful methods to detect humans in static images and then continue to methods which rely on motion information as well as the information from static images. Finally we will review some methods to handle occlusion in single frame and video sequences. The literature for human detection is vast and different features for human detection were introduced. In this survey we mainly focus on the methods which are widely used in the computer vision community. For a more extensive review of the human detection methods the reader can look at Dollár et al. [13].

2.2 Human Detection Based on Appearance

One of the most popular and influential methods to detect humans is introduced in Dalal and Triggs [8]. This paper introduced a new and powerful feature, Histogram of Oriented Gradient (HOG), for human detection. The intuition behind this method is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the responding gradients or edge positions. To do so they divide the window of interest into cells and for each cell they generate a histogram of gradient directions or edge orientations. Then they concatenate the histograms of different cells and generate a complete feature vector and finally they feed this feature vector to a SVM classifier to do the detection. The authors investigated the effect of different parameters on the performance of the whole system. The

complete proposed system can be seen in Fig. 2.1.

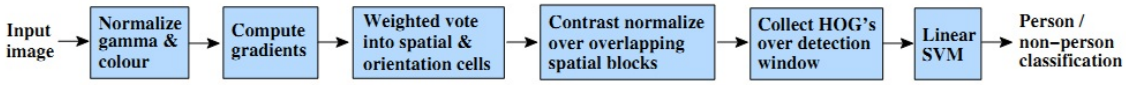

Figure 2.1: HOG feature extraction and pedestrian detection system. Figure taken from [8].

The first block in the detection chain is the Gamma Color Normalization block where the authors chose square root gamma compression for normalizing the color of the window. In order to have a good feature the authors divided each input image to some overlapping blocks (16×16 pixels blocks) and then they divided each block to some cells (8×8 pixels cells), the normalization block is applied to the blocks. However because of other normalization blocks in the chain the effect of this block on the whole performance is negligible.

The performance of the system is very sensitive to the calculation of the gradients, it turned out that the simplest method to calculate the gradients is the most effective one. The gradient calculation block is realized using simple 1-D point derivatives without any pre-smoothing on the input image. For color images the gradients are calculated on each color channel separately and then for each pixel the channel with largest norm is picked as the pixel's gradient vector.

The next step is to define the feature vector using the calculated gradients. In each cell, each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centred on it, then these votes are accumulated into orientation bins in the cell area to generate the feature vector. The bins are evenly distributed between 0° and 180° and the weight of the vote is defined as the magnitude of the gradient at the point.

The last step for generating feature is to define the blocks and how to generate the cells in each block. The authors checked the two common schemes to define blocks and divide them (rectangle and circular blocks) and found out that the rectangular blocks with spatial overlap would result in good performance. Each block's feature vector is then normalized using the $L_2 norm$, $v \rightarrow v/\sqrt{||v||_2^2 + \epsilon^2}$. After this step the feature vector is ready for the classifier to do the classification.

The classifier used in this paper is the SVM classifier. They investigated the effect

of different kernels in the performance of the whole system, although the Gaussian kernel generates the best results but the computation efficiency of the linear kernel makes this kernel more suitable for the task.

HOG detection system achieves near perfect detection on the MIT pedestrian database [21] so the authors introduced a new and more challenging dataset to test their method, "INRIA" [8], and they outperform the other methods available in that time. The results can be seen in Fig. 2.2.

We used the HOG pedestrian detector as one of our baselines and also integrated it with our system to help our system detect non-moving pedestrians. An extension of simple

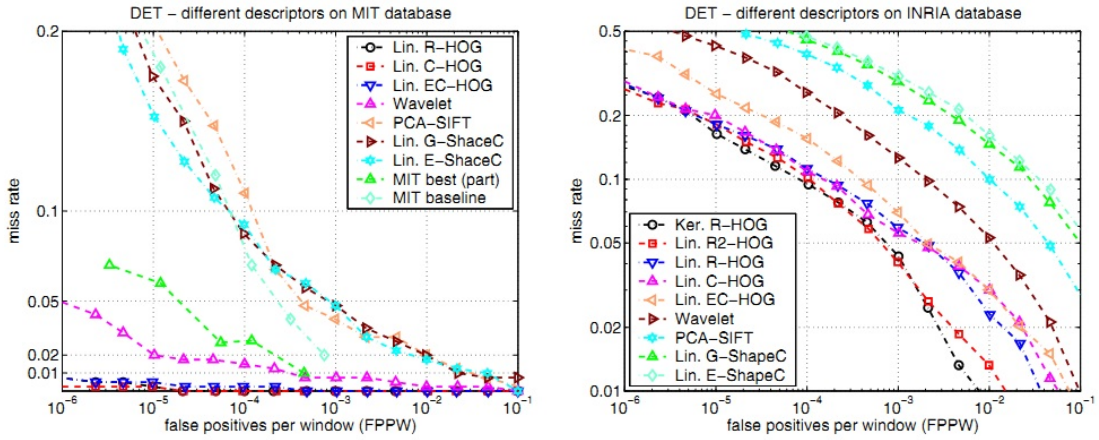

Figure 2.2: The performance of HOG versus other detectors. Figure taken from [8].

HOG detection system is introduced in Felzenszwalb et al. [15] where the authors used HOG features and introduced a set of latent variables to do the detection. These latent variables represent the formation of different parts of the body and with the scoring scheme used in this paper, the authors achieved better detection performance over the original HOG method.

The proposed method uses a pyramid scheme to do the detection similar to the Dalal et al. [8]. They modelled human body with a star like structure in which a node for each body part and to model the interaction between different parts they added an edge between two nodes, the star term means there is no loop in the final structure. In their proposed star model, a root detection is done in coarse level which covers the object and then in the finer levels they used some local filters which covers smaller parts of the object. These smaller

parts can be interpreted as different body parts. If we could detect and locate the body parts accurately then we could detects humans more precisely. Unfortunately this is not the case here so the authors treated the part detections as latent variables which means that these detection may be inaccurate but still useful for human detection. An example of the detection of these parts and its meaning is given in Fig. 2.3. As it can be seen there is no need to exactly localize the body parts and it still has a good detection.

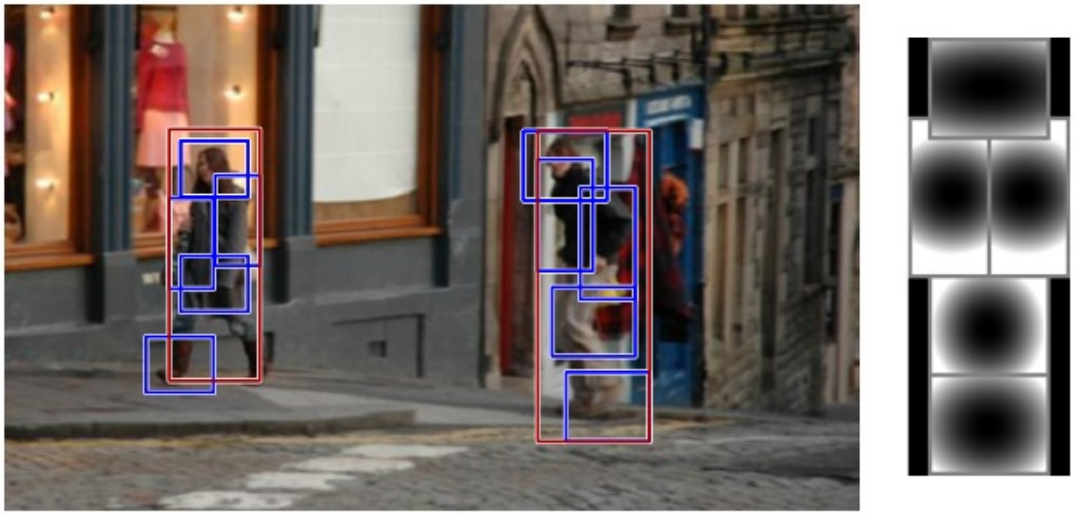

Figure 2.3: An example of human detector and the body part detection. Figure taken from [15].

To solve the problem properly the authors proposed a latent SVM framework which can solve deformable part based model detection efficiently. There is a vast literature on human detection using different sets of features and representations of body parts but the the above mentioned methods are widely accepted and used in many other systems and proved their effectiveness in real world problems.

2.3 Detecting Pedestrians Using Appearance and Motion Information

The simplest idea to use motion is to extract the background and then calculate the difference between background and the current frame to extract the objects which are added

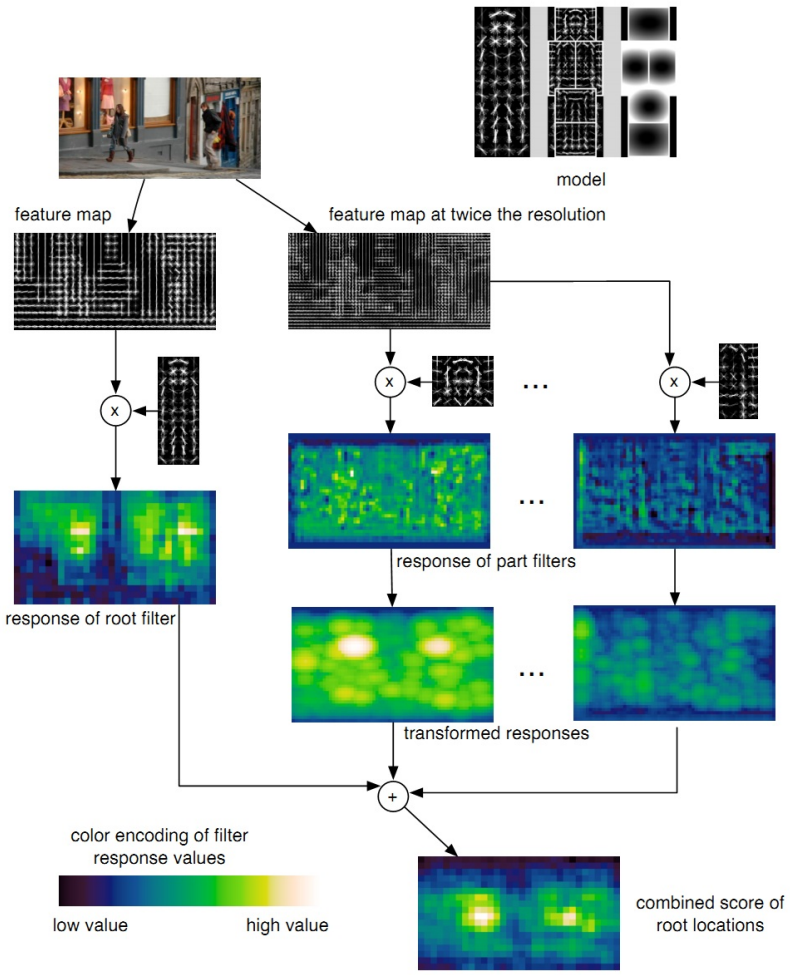

Figure 2.4: Body part detection. Figure taken from [15].

to the scene and highlight them as the region of interest. Haritaoglu et.al [17] proposed a method in which they extract background in the frames where there is no human in the scene and then use the information about the background, min value and max value of each pixel in the background, to determine if a pixel belongs to foreground or not. Then they clean the foreground using morphological operations and do the detection and the action recognition on the extracted blob. In their work they assume that they have scenes with only background in it but if the camera is fixed the background can be extracted by calculating mean or median of a pixel over a period of time. Although this method used the

motion information to highlight the object of interest but it doesn't differ between the human motion or other object's motion. In the next few parts we will explain methods which are trying to capture the difference between the motion of a human and other objects in the scene.

One of the first successful usage of motion patterns to do the human detection is proposed in Viola et al. [24], where the authors used the Haar wavelet to extract appearance feature and used a simple motion descriptor to extract the motion pattern of the subject. They also used a cascade structure to improve the detection results.

The most important piece of this work is the motion descriptor, instead of using optical flow, which is very common to describe motion, they used a simple descriptor using shifted frames and Sum of the Absolute Difference to estimate the displacement between frames. The formulation of their motion descriptor is as follow,

$$\begin{aligned} \Delta &= abs(I_t - I_{t+1}) \\ U &= abs(I_t - I_{t+1} \uparrow) \\ L &= abs(I_t - I_{t+1} \leftarrow) \\ R &= abs(I_t - I_{t+1} \rightarrow) \\ D &= abs(I_t - I_{t+1} \downarrow) \end{aligned} \tag{2.1}$$

Where I_t and I_{t+1} are two frames of the video and $\uparrow, \downarrow, \leftarrow, \rightarrow$ are the shift operators which shift their inputs by one pixel in their direction. Using these operators the authors introduced 3 types of filters which each one captures a specific kind of motion. The most important filter is a set of Haar wavelet filters, the very same filters used for extracting appearance features. An example of these filters ans motion displacement is shown in Fig. 2.5.

Although the filtering is done using integral image but to speed up the process more the authors proposed to use a cascade structure for their detector, they trained the cascade structure and the detector using AdaBoost learning structure. AdaBoost scheme is very useful when using wavelets as the input features because AdaBoost can select most discriminative features and ignore the rest.

Inspired by Viola et al. [24], the authors of [8], Dalal and Triggs, proposed an extension to their work where they added motion information to their detector to improve the results, Histogram of Optical Flow (HOF). This new information captures the relative motion of different body parts and add it to the classifier to improve the results of the detection. The

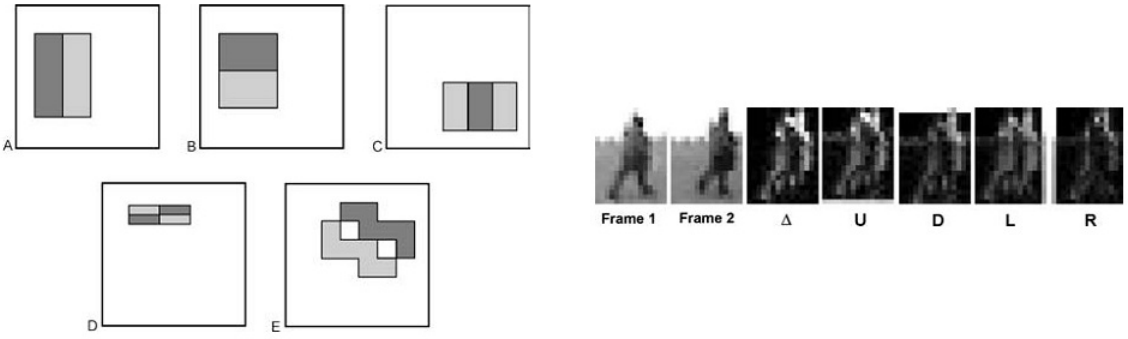

Figure 2.5: Haar wavelet filters and displacement images. Figure taken from [24].

main advantage of this work over Viola et.al [24] is that the motion representation here is more robust to camera motion and background motions. An overview of the proposed method can be seen in Fig. 2.6. The motion descriptor used in this paper is based on flow

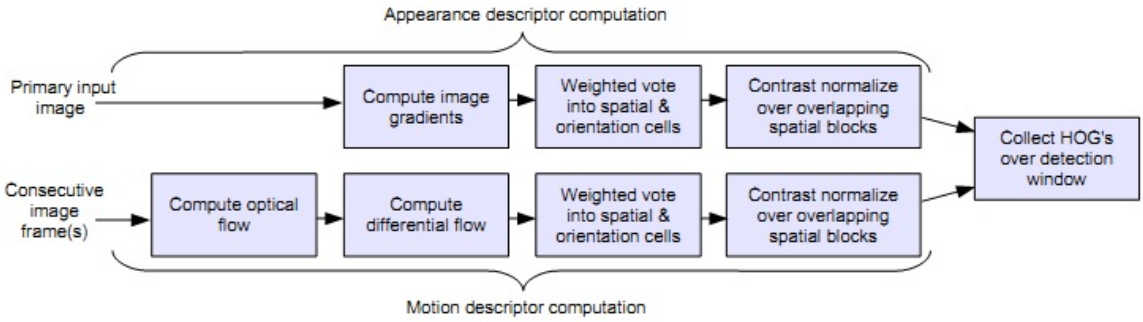

Figure 2.6: Feature extraction and detection process. Figure taken from [9].

differential which can remove camera rotation and slow movement of the camera, because these movements results in smooth variation in optical flow. The independent motions are the largest at motion boundaries so this feature can simply highlight the outline of the subject but the internal dynamics of the subject is very important and differential optical flow can capture the relative motions of the limbs which are useful for the classification which would be a very informative addition to the system.

In order to use optical flow efficiently the authors proposed many different filters to compute motion patterns, we will explain the filters in the following part.

IMHdiff is the first filter, it takes derivatives and uses the (I_x^x, I_x^y) and (I_y^x, I_y^y) to create

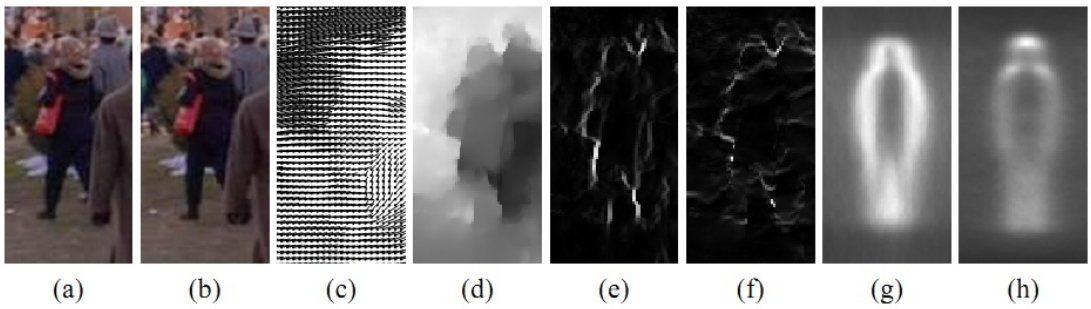

Figure 2.7: Illustration of the HOF descriptor. (a,b) Reference images at time t and t+1. (c,d) Computed optical flow, and flow magnitude showing motion boundaries. (e,f) Gradient magnitude of flow field I^x, I^y for image pair (a,b). (g,h) Average HOF descriptor over all training images for flow fieldI^x, I^y. Figure taken from [9].

two relative-flow-direction based oriented histograms, where I^x and I^y are the magnitude of the optical flow in x and y directions and (I^x_x, I^x_y) (I^y_x, I^y_y) are the gradient of those optical flows in the x and y directions,an example of (I^x_x, I^x_y) and (I^x_y, I^y_y) can be seen in Fig. 2.7.

IMHcd uses 3×3 blocks of cells, in each of the 8 outer cells computing flow differences for each pixel relative to the corresponding pixel in the central cell and histogramming to give an orientation histogram.

IMHmd is similar to IMHcd the only difference is that instead of doing the calculation on each pixel of the cell the calculation are done on the average of the cells and results in 9 histograms.

IMHwd is similar to IMHcd except that instead of using non-central differences, Haar wavelet operators are used as the filter.

In addition to the mentioned filters the authors utilized the motion descriptor used in Viola et al. [24] and named it **ST DIFF**.

One of the first steps in this method is to compute the optical flow. It turns out the calculation method can significantly affect the performance of the whole system. Different experiments showed that any smoothing before calculation of the optical flow can reduce the performance of the system. They proposed to use the simplest method to calculate the optical flow without any smoothing and in a coarse to fine manner. They calculate the initial flow in the coarse level of pyramid and refine the results in the finer levels.

The methods proposed in Viola et al. and Dalal et al. [24, 9] are the basis of most of the

other detection systems which use motion features as their input feature. We will review another paper on using motion features to do the detection in the next section and then explore the problem of occlusion.

2.4 People Detection using Tracking

Another interesting trend in human detection is to track an object in different frames and then use the highest score of a shape based detector in the frames to decide whether an object is a human or not.

Garcia-Martin et al. [16] introduced a two level detection where an initial detection is done using appearance and motion model then they do a tracking on the candidate and then update their detection on the estimated position of the bounding box in the next few frames to get a more reliable detection, an overview of their system can be seen on Fig. 2.8.

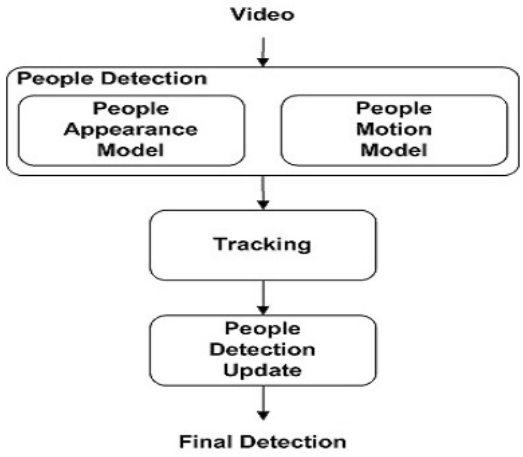

Figure 2.8: System overview. Figure taken from [16].

For their appearance model the authors used Implicit Shape Model (ISM) by Leibe and Schiele [18], this model consists of a codebook of local appearances that are the prototype of the objects. Each prototype has its own probability distribution which is used for classification, to generate the codebook a clustering is applied on the input space and divide it into K classes.

To generate a motion model the authors used the idea of ISM and applied it to motion

information. To build the motion model a variation of SIFT point detector and descriptor is used, this variation can capture the interest points based on their appearances and local motion. This descriptor is called motion SIFT or MoSIFT and is generated using SIFT, the generation of this feature has 3 steps. First SIFT is applied to the input frame and the interest points are extracted then optical flow is calculated around these points and finally the MoSIFT is generated, an example of this feature is shown in Fig. 2.9.

Figure 2.9: SIFT (left) and MoSIFT (right) interest points. Yellow circles indicate interest points and their scales, red arrows indicate the dominant motion orientation. Figure taken from [16].

After calculating MoSIFT they build the codebook which tries to categorize the motion patterns of the subjects. To do testing on an input image, first SIFT and MoSIFT points are extracted and then the motion patterns are extracted, then the distance between these patterns and different clusters are calculated. Each cluster casts votes for hypothetical positions of the person center according to the learned spatial distribution P_C. These centers are considered as hypothesises, and the overlapping hypothesises are simplified to the highest score one. Results of this method can be seen in Fig. 2.10.

In Felzenszwalb et al. [15], the authors tried to improve the HOG by detecting different body parts. Similarly the natural extension over the idea presented in Garcia-Martin et al. [16] is to detect and track body parts and improve the detection results.

Andriluka et al. [2] introduced a method in which they try to detect limbs and estimate the human pose to detect the humans and then track each limb and improve their detection using this tracking information. Their method is also capable of handling occlusion since

Figure 2.10: Detection process examples. Voting space (black lines), center hypotheses (green points), hypotheses (red rectangles) and final hypothesis (green rectangles). Figure taken from [16].

they model the humans as a set of limbs and the problem of having some occluded limbs can be handled by detecting other visible limbs.

They introduced a pictorial model to model the body. In order to cope with the variations of the body shape during walking the authors introduced auxiliary state variables that represent the articulation state which stands for different phases in the walking cycle of a person. Knowing the walking phase the pictorial model can be modelled as a star model so the inference can be done efficiently using dynamic programming.

The proposed method detects humans from the information of the single frame but to add the motion information to improve the performance they used a chain-like structure to capture the dependencies between consequent poses. This chain is modelled using hierarchical Gaussian Process Latent Variable Model (hGPLVM) where the latent variables are the pose dynamics an example of their method can be seen in Fig. 2.11.

Another important line of research in pedestrian detection is to handle occlusions. Occlusions frequently happen in real world scenarios, these occlusions make the task of detection very hard because some parts of the body would be missing so the total score of the whole body detectors may reduce significantly and as a result the detector fails. There are many available methods to handle occlusion. Here we will describe some of these papers.

Figure 2.11: An example of the detection. Figure taken from [2].

2.5 Detection of Multiple and Partially Occluded Humans

One of the methods to handle occlusion is introduced by Wu et al. [26]. In this work the authors tried to detect three parts of the body: head-shoulder, torso and legs, and combine these detections to detect humans. As the feature they introduced an edgelet feature which is calculated on the edge of the silhouette of the subject. Calculating the feature on silhouettes reduces the effect of different clothing on detection and also their proposed method is capable of handling some variation in view point.

The system has some assumptions about the condition for the detection, first the camera is looking down to ground and the scale of people closer to the camera is bigger and the head of the pedestrians are always visible. Using these assumptions the system builds an occupancy map which captures the relative positions of possible candidates based on the responses from whole body detector and head detector.

Having the occupancy map, the system tries to estimate the possible occlusions in the scene and based on those possible occlusions and responses of different part detectors the system refines its detection and update the occupancy map, Fig. 2.12 shows an example of these steps.

The system models the problem using Maximum a Posteriori Estimation (MAP), following the notation in [26] the system models the observation as follows:

$$p(I|S) = p(Z|\tilde{S}) = p(Z^{FB}|P^{FB})p(Z^{HS}|P^{HS})p(Z^{T}|P^{T})p(Z^{L}|P^{L}) \tag{2.2}$$

where $\tilde{S} = \left\{ \left\{P_i^{FB}\right\}_{i=1}^{m^{FB}}, \left\{P_i^{HS}\right\}_{i=1}^{m^{HS}}, \left\{P_i^{T}\right\}_{i=1}^{m^{T}}, \left\{P_i^{L}\right\}_{i=1}^{m^{L}} \right\}$, is the reduced version of S by removing all occluded body parts and superscripts FB, HS, T and L stand for full-body, head-shoulder, torso and legs respectively and $m^{FB}, m^{HS}, m^{T}, m^{L}$ are the number of visible parts, $P = FB, HS, T, L$. In order to decide if a part is occluded or not the authors

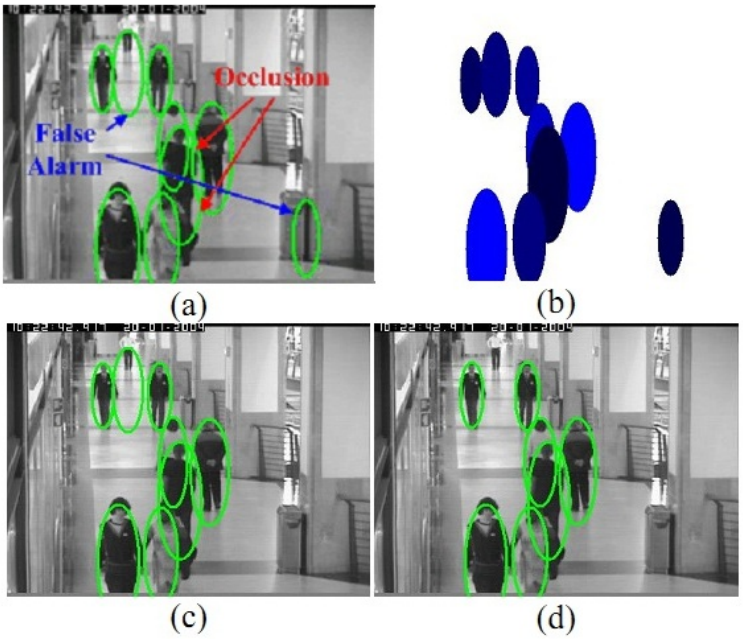

Figure 2.12: Search for the best interpretation of the image: a) initial state; b) occupancy map of the initial state; c) an intermediate state; and d) final state. Figure taken from [26].

calculated the ratio of the visible part to the total area of the part and if it is above a threshold then it is considered to be visible.

Most of the presented papers worked in less crowded scenes and when the number of people in the image increases they fail to detect pedestrians accurately. There are two main problems in crowded scenes. First the computation cost usually increases exponentially as the number of the subjects in the scene increases. The other problem with crowded scenes is that with routine sliding window scheme there would be many detections around a single pedestrian and these many detections usually are removed using a non-maxima suppression (NMS) to generate a single detection. But in a crowded scene a NMS can remove many true detections and degrade the performance of the system. The proposed method in Yan et al. [27] can handle multiple detections without using NMS. They used Latent Rank SVM which has a reasonable computation cost.

The model tries to combine appearance cues and spatial information. They propose a probabilistic model to explore the relationship between two clues and solve the model using MAP. They used Felzenszwalb et al. [15] part based model as their appearance model and the spatial model is used to describe the interactions of the subjects the interaction model

uses the scale of the candidates and their relative positions to capture the interaction of the candidates.

In order to handle the occlusion, the authors proposed a method to find out what kind of occlusion happened. Based on the type of occlusion they re-weight the Felzenszwalb et al. [15] part based model weights and try to put more weight on visible parts while suppressing the effect of the occluded parts on the total score of the detector. The type of the occlusion is not available and it is not the output of the final system so it will be treated as a latent variable.

Unfortunately the type of occlusion is usually not provided in the training sets so an EM-like approach is employed to estimate the occlusion type as well as the true label of input window. In order to have a good initialization a clustering is applied to the data and divide the input space to K clusters. Instead of solving the latent model using the common latent SVM model they formulated the problem as a ranking problem where the true label should get a higher score while the other getting lower score.

An important contribution of this paper is that they utilized the spatial information of the image in the detection so their final system doesn't need any NMS and can work in crowded scene, an example of the detection in a crowded scene is provided in Fig. 2.13.

Figure 2.13: An example of detection in a crowded scene. Figure taken from [27].

While Yan et al. [27] tried to capture different types of occlusions using clustering, Shu et al. [23] proposed another system where they limit the types of occlusion. Then by assuming they know the type of the occlusion they tried to solve the detection problem in the presence of occlusion.

Like many other works done on pedestrian detection, this paper introduced a discriminative model for pedestrian detection and used a part based model to do the detection. They considered 3 predefined types for the appearance of the visible parts, namely, head

only, upper body parts and all body parts.

In order to decide which type of occlusion happened Yan et al. [27] choose a subset of parts $P = \{p_0...p_n\}$ which maximize the following score for a given bounding box at the position of (x,y_0),

$$score(x_0, y_0) = b + argmax_{S_m} \frac{1}{|S_m|} \times \sum_{i \in S_m} \frac{1}{1 + exp(A(p_i).s(p_i) + B(p_i))} \tag{2.3}$$

where $s(p_i)$ is the score of the part i from the part detector and A, B are learned using a sigmoid fitting approach. $|S_m|$ is the set cardinality. This formulation is very similar to other works done on occlusion handling but they tried to add some information from tracking to handle the occlusion better.

The first important observation is that the occlusions are highly correlated in the adjacent frames of video, so it is reasonable to share the information about the occlusion state of a subject between frames. The authors proposed an occlusion prediction method and used it in their tracking algorithm. In their tracker they have a person classifier to detect individuals using an specific detector for every person in scene which learns the appearance of the subject in an online manner. Using the information about the occlusion in the frames they update the appearance model only on visible parts. Also, they use their occlusion predictor to focus on more visible parts in the next frames to do the detection so they would have a more reliable tracking. An example of their detection and tracking can be seen in Fig. 2.14.

2.6 Oriented Energy Features

Events in a video sequence will generate diverse structures in the spatiotemporal domain. For instance, a textured, stationary object produces a much different signature in image space-time than if the same object were moving. One method of capturing the spatiotemporal characteristics of a video sequence is through the use of oriented energies Adelson et al. [1]. These energies are derived using the filter responses of orientation selective bandpass filters when they are convolved with the spatiotemporal volume produced by a video stream. Responses of filters that are oriented parallel to the image plane are indicative of the spatial pattern of observed surfaces and objects (e.g., spatial texture) whereas, orientations that extend into the temporal dimension capture dynamic aspects (e.g., velocity and flicker).

Figure 2.14: An example of detection and tracking in a crowded scene. Figure taken from [23].

Cannons and Wildes [5] used these Spatiotemporal Oriented Energies (SOE) to track objects. The idea behind used approach is that energies computed at orientations which span the space-time domain can provide an extremely rich description of a target for visual tracking. Multiscale processing is important in this tracker, as coarse scales capture gross spatial pattern and overall target motion while finer scales capture detailed spatial pattern and motion of individual parts (e.g., limbs). With regard to dynamic aspects, simple motion is captured (orientation along a single spatiotemporal diagonal) as well as more complex phenomena, e.g., multiple juxtaposed motions as limbs cross (multiple orientations in a spatiotemporal region). By encompassing both spatial and temporal target characteristics in an integrated fashion, tracking is supported in the presence of significant clutter. In addition to robustness against occlusions this tracker can be made invariant to local image contrast to support tracking in the presence of substantial illumination changes which makes this method an excellent choice for a system which needs to work in uncontrolled environment.

To calculate the SOEs, filtering was performed using broadly tuned, steerable, separable

filters based on the second derivative of a Gaussian, G_2, and their corresponding Hilbert transforms H_2, with responses point-wise rectified (squared) and summed. Filtering was executed across $\theta = (\eta, \xi)$, 3D orientations ((η, ξ) specifying polar angles) and σ scales using a Gaussian pyramid formulation. This Gaussian pyramid approach allows for efficient analysis of the space-time structure across multiple scales. Hence. a measure of local energy, e, can be captured according to,

$$e(X; \theta, \sigma) = [G_2(\theta, \sigma) \star I(X)]^2 + [H_2(\theta, \sigma) \star I(X)]^2 \tag{2.4}$$

where $X = (x, y, t)$ corresponds to spatiotemporal image coordinates, I is the image sequence, and $\star$ denotes convolution. This initial measure of local,local with respect to θ and *sigma*, energy is dependent on image contrast. To attain a purer measure of the relative contribution of different orientation irrespective of local contrast, $e(x; \theta, \sigma)$ is normalized as

$$\hat{e}(x; \theta, \sigma) = \frac{e(x; \theta, \sigma)}{\sum_{\tilde{\sigma}} \sum_{\tilde{\theta}} (x; \tilde{\theta}, \tilde{\sigma}) + \epsilon} \tag{2.5}$$

where ϵ is a bias term to avoid instabilities when the energy content is small and the summations in the denominator cover all scale and orientation combinations.

Fig 2.15 displays a subset of the energies that are computed for a single frame of a MERL traffic sequence, Brand et al. [4]. Here, there is a white car moving to the left near the center of the frame. Notice how the energy channel that is tuned for leftward motion is very effective at distinguishing this car from the static background. Consideration of the channel tuned for horizontal structure shows how it captures the overall orientation structure of the white car. In contrast. while the channel tuned for vertical textures captures the outline of the crosswalks, it shows little response to the car. as it is largely devoid of vertical structure at the scales considered. Finally, note how the energies become more diffuse and capture more gross structure at the coarser scale.

Given that the tracking problem being considered, the goal is to locate the target's position as precisely as possible. However, as seen in Fig 2.15, the energies computed at coarser scales are diffuse due to the downsampling/upsampling that is employed in pyramid processing. Coarse energies are important because they provide information regarding the target's gross shape and motion, but a method is required to improve their localization for accurate tracking. To that end, a set of weights are applied to the normalized energies of Eq. 2.6 according to

$$\hat{E}(x; \theta, \sigma) = \hat{e}(x; \theta, \sigma) b(x; \theta) \tag{2.6}$$

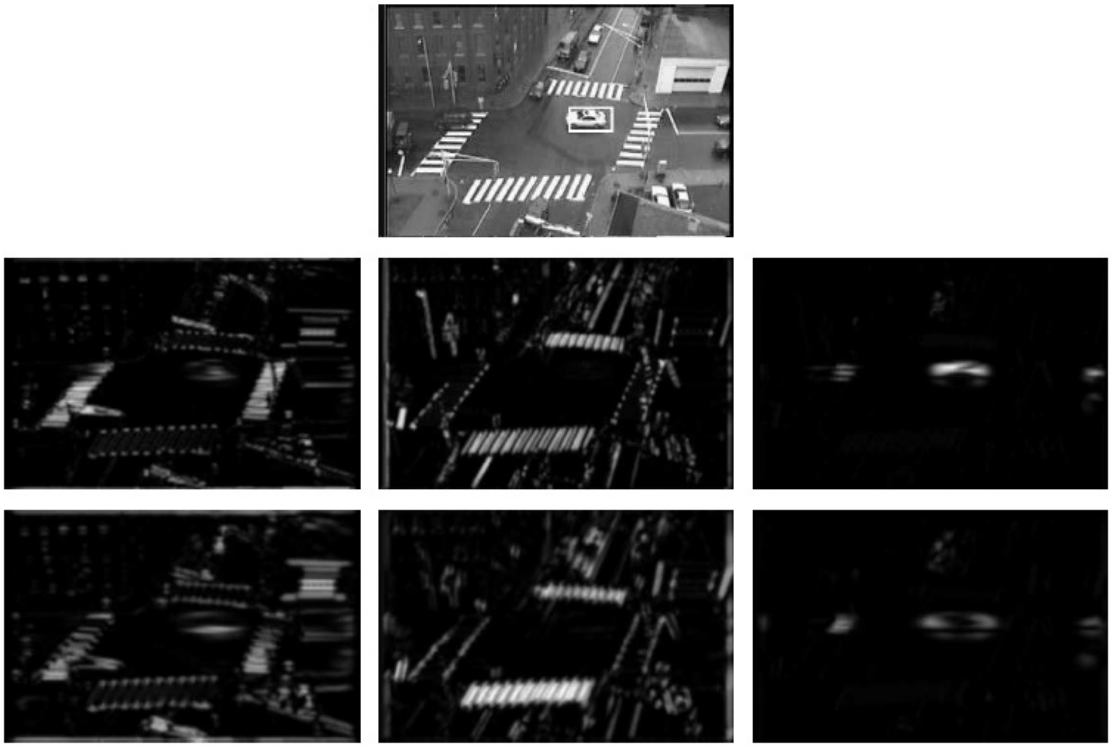

Figure 2.15: Frame 29 of the MERL traffic video sequence with select corresponding energy channels. Finer and coarser scales are shown in rows two and three, resp. From left to right, the energy channels roughly correspond to horizontal structure, vertical structure, and leftward motion.

where b is pixel-wise weighting factor for a particular orientation channel, θ. The weighting factors for a specific orientation are computed by integrating the energies across all scales and applying a threshold, T_θ, according to

$$b(x;\theta) = \sum_{\tilde{\sigma}} \hat{e}(x;\theta,\tilde{\sigma}) > T_\theta \tag{2.7}$$

When computing the weights, summing across scales allows the better localized fine scales to sharpen the coarse scales, while the coarse scales help to smooth the responses of the fine scales. Furthermore, by calculating weights separately for each orientation, being prejudiced toward any particular type of oriented structure (e.g., static vs. dynamic) is avoided.

The proposed oriented energy feature set has two important advantages. First, normalized energy, as defined by Eq. 2.6 and Eq. 2.7, captures local spatiotemporal structure at a particular orientation and scale with a degree of robustness to scene illumination. By virtue

of the bandpass filtering, Eq. 2.6, invariance will be had to changes that are manifest in the image as additive offsets to image brightness, by virtue of the normalization, Eq. 2.7, invariance will be had to changes that are manifest in the image as multiplicative offsets. Second, the calculation of the defined normalized oriented energies requires nothing more than 3D separable convolution and pointwise nonlinear operations, and is thereby amenable to compact and efficient implementation, Derpanis et al. [10].

As defined, oriented energies provide local characterisation of image structure. Therefore, the pointwise measurements can be aggregated over target support to provide region-based descriptors (e.g., in conjunction with mean shift tracking [7]). Also, the information about the image structure can be used to describe the shape of the objects in image and do object detection.

Since the proposed tracker is a inspired by mean shift tracker [7], authors collapsed the spatial information in energy measurements and represent the target as a histogram. The target histogram is constructed using the energies of Eq. 2.8. Each histogram bin corresponds to the weighted energy content of the target at a particular scale and orientation. Hence, the entire histogram displays the weighted energy of the target across all scales and orientations. The energy histograms are created in a different fashion than the color histograms seen in many mean shift algorithms. Unlike most of mean shift tracking algorithms, when computing energy-based histograms, each target pixel affects every bin in the histogram. This histogram is calculated in the first frame of the video sequence by,

$$\hat{q_u} = C \sum_{i=1}^{n} k(||x_i^*||^2) \hat{E}(x_i^*; \phi_u) \tag{2.8}$$

where k is the profile of the tracking kernel, C is a normalization constant to ensure the histogram sums to unity, $x_i^* = (x^*, y^*)$ is a single target pixel at some temporal instant, i ranges so that x_i^* covers the template support, and ϕ_u is the scale and orientation combination which corresponds to bin u of the histogram.

When tracking a target between frames, it may be necessary to evaluate several target candidates before a final, optimal target position is found for the current frame. The histograms for the target candidates are evaluated using

$$\hat{p_u}(y) = C_h \sum_{i=1}^{n_h} k\left(||\frac{y - x_i^*}{h}||^2\right) \hat{E}(x_i^*; \phi_u) \tag{2.9}$$

where y is the center of the target candidate's tracking window. h is the bandwidth of the

tracking kernel and i ranges so that x_i^* covers the candidate support. The kernel bandwidth allows for scale changes of the target throughout the video sequence.

A sample energy histogram for the target region shown in Fig. 2.15 (represented by the white box) is shown in Fig. 2.16. The bin corresponding most closely to leftward motion at finest scale (bin 5) has by far the most energy. The next two high energy counts are found in bins 2 and 9 which are tuned to combinations of dynamic and static structure, with an emphasis on leftward motion and spatial orientation similar to that of the target. The overall horizontal structure of the car is captured by the energy in bins 1 and 4. In contrast, bins 3 and 6, which roughly represent static, vertical structure, do not have strong responses, given the nature of the car target. The histogram also shows that the oriented energies for the highest frequency structures have the strongest response, as the target is fairly small and dominated by relatively finer scale structure.

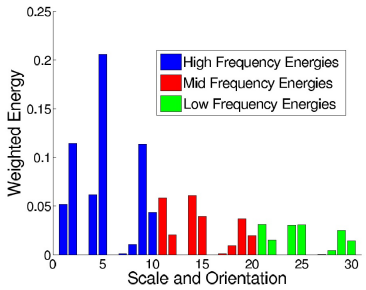

Figure 2.16: Oriented energy histogram for the target region in Fig. 2.15.

Chapter 3

Human detection using latent tracking

In this chapter we are going to explain the model for human detection as well as datasets and methods used to train and test the proposed pedestrian detection method.

In the first part we will explain the model and the formulation of our system and in the following parts we will explain the datasets and the experiment setups and in the final section we will see the results and do the discussion.

The main idea behind the proposed detection system is to use the motion patterns to improve detection results and adding robustness to the detection system. We are modelling the motion patterns by tracking different parts of the bounding box and using its track and its relative displacements with respect to its neighbours. This model tries to capture the common motion patterns of the pedestrians, for instance the motions of feet and the relative motions of hands and feet. The idea of the total system can be seen in Fig. 3.1.

3.1 Model

For our model we start from a bounding box and calculate a global feature on the whole bounding box (HOG feature and SOEs). Then we divide the bounding box of the first frame into some sub-rectangles and then track each sub-rectangle in the next few frames and then use these tracks and the relations between tracks to do the detection. We define different types of the weights for each track and also we have different set of weights for

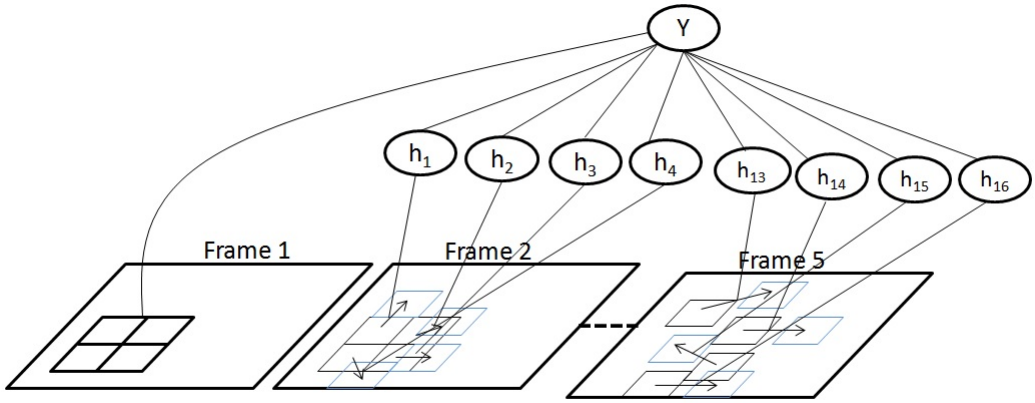

Figure 3.1: Overview of the whole system.

the interactions between tracks. In this thesis we divide the initial bounding box into 6 sub rectangles and track each rectangle separately. These tracks are considered as latent variables and then used to detect pedestrians. These tracklets are very meaningful and they can help detecting pedestrians reliably, these tracklets will help detecting rigid moving objects such as cars and motorbikes since they don't have any rhythmic motion or any other kind of patterned movements except for simple displacements while the human body parts show some kind of rhythmic (patterned) motion.

Since the output of a tracker is a continuous variable and it can get any number in the search region. We quantize the output of the tracker, round to the nearest multiple of 5, to reduce the number of possible values for the latent variables. At the end of this step we have a set of latent variables which show the track of each body part in the frames. We will use this tracklets to do the detection.

We defined different types of relations between our latent variables our model has both unary and pairwise terms, we have a unary term for each latent variable but for the pairwise term it is way more complicated to generate a term for each possible pair so we restricted the pairwise relations to the adjacent blocks, the pairwise relations can be seen in the Fig. 3.2.

Given an input bounding box, we use the latent SVM formulation to solve our problem. Here is the formulation of our model:

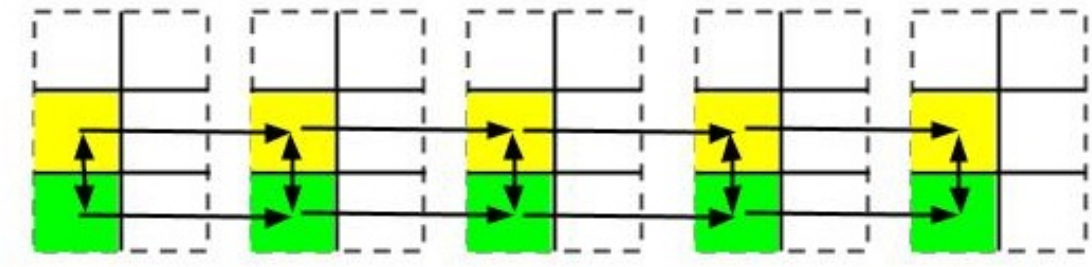

Figure 3.2: The pairwise relation between latent variables.

$$
\begin{aligned}
f_w(x,h) &= w^T\Psi(x,h) && (3.1)\\
&= w_0^T\phi_0(y,x_0) + \sum_j w_1^T\phi_1(h_j) + \sum_j w_2^T\phi_2(h_j,x_{j,b}) \\
&+ \sum_j\sum_i w_3^T\phi_3(h_i,h_j) && (3.2)\\
w_1^T\phi_1(h_j) &= \sum_{b\in H} w_{1,b}1(h_j == b) && (3.3)\\
w_2^T\phi_2(h_j,x_j) &= \sum_{b\in H} w_{2,b}^T 1(h_j == b)x_{j,b} && (3.4)\\
w_3^T\phi_3(h_i,h_j) &= \sum_{b\in H}\sum_{c\in H} w_{3,b,c}1(h_i == b)1(h_j == c) && (3.5)
\end{aligned}
$$

where $f_w(x,h)$ is the score of input bounding box, the higher this score is the more chance that the input bounding box is a human. x is our feature vector and h is our quantized displacement vector for each part, $h \in \{(0,0),(0,5),...,(0,N{*}5),(5,0),(5,5)...,(N{*}5,N{*}5)\}$. Function 1(.) is the indicator function where it equals one when the inner term is true and 0 otherwise. The term $w_0^T\phi_0(y,x_0)$ models our global feature which we used HOG and a HOG like feature which is defined on SOEs, this term can give us some assurance that our system in the worse case (where our new features are completely random and have no correlation with detection) would perform like HOG and hopefully our method and features will improve over HOG. We are using linear SVM for this model which is very fast and commonly used for detection systems, as a result the $\phi_0(y,x_0)$ would be the HOG features of the original image concatenated with the HOG features calculated from SOE channels.

$\phi_1(y, h_j)$ models the unary term which is the relation between the our latent variables (displacement of each part in each frame) and the detection results, in its simplest form we can say that this model somehow checks if the speed of each body segment is similar to the speed of a human or not, we have a constant frame rate so the displacements between frames can be seen as the speed of that part. For example the speed or displacement of the head part should not be very high because there is an upper limit for the speed of a pedestrian which is different from the speed of other moving objects in the scene.

$\phi_2(h_j, x_j)$ models the compatibility between the displacement of size h_j and the its corresponding image feature x_j. x_j is the feature vector extracted from the images we are doing the tracking on them. This term ensures that the model does track different parts. x_j is generated similar to the method introduced in [5]. It contains information about shape and speed of the tracked part.

Finally, $\phi_3(y, h_i, h_j)$ models the relationship between our latent variables, based on our model, Fig. 3.2, it basically checks two important relations, the first one is the relation between the displacements(speeds) of one body part between frames which we expect to be similar and not very different, seconds it model the relative displacements of two adjacent body parts. This term can capture motion patterns which are useful for human detection, for instance it captures the relative displacement of hands and feet which is a very discriminative feature for detecting walking pedestrians.

In order to have a reasonable latent variable we define the latents as the displacement of a patch from its position in the previous frame. In theory these displacements can be any value and if we don't limit these values the inference and finding the maximum would be intractable so we will limit our displacement values to a limited set of numbers.

In the inference part of algorithm we have to find proper values for latent variables based on the possible output value and the values for other latent variables. Solving the exact inference for this problem is very time consuming because we have to do a tracking with all the possible weights and this is a computationally expensive task, so we use a beam search based approach to solve the inference and maximize the goal function. The scheme of this beam search is shown in Fig. 3.3. Beam search is a search method from greedy optimization family so the computation time expands linearly by adding more frames. Since we are limiting the search space the beam search solution might not be the best solution but the experience results showed that with a proper choice for limiting the search space this inaccuracy wouldn't be significant and it wouldn't affect the performance of the detection

system. In the used beam search we save the top 25 candidates and each frame we do search on each top 25 candidates and after the search we cut the solutions to the top 25 based on their score.

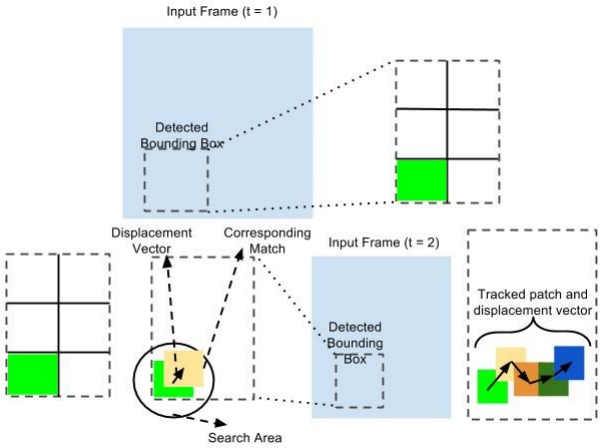

Figure 3.3: Beam search scheme.

3.2 Learning

For finding the weights of the system we used Non convex Regularized Bundle Methods (NRBM) package of Do and Artières [11] to optimize the goal function this package is guaranteed to converge to a local optima, we used a linear kernel for our goal function so this package would work efficiently for finding the weights.

We use $y^* = argmax_y E(x, y; w)$ as the predicted label of x where y can be ± 1. Given $(x^1, y^1), (x^2, y^2), ..., (x^n, y^n)$, the set of training data, we aim to find parameters that the sign of score x^i and y^i represent the class of the X^i. Similar to Felzenszwalb et al. [15] we formulate the training criteria in the Max-Margin framework. We set w by:

$$min_{w,\xi^i} \frac{\lambda}{2}||w||^2 + \sum_{i=1}^{N} \xi^i \tag{3.6}$$

$$\text{s.t. } y_i(E(x^i; w) + b) \geq 1 - \xi^i, \xi^i \geq max(0, 1 - y_i(E(x^i; w) + b))$$

where λ is a tradeoff constant and is determined using cross validation in different experiments. These constraints would force the score of positive samples to be positive and greater

than one while forcing the score for negative samples to be negative and less than -1.

Chapter 4

Experiments and results

In this chapter we will explain the used datasets and their characteristics as well as the experiments we ran on them and the parameters used for the experiments. Since most of the current datasets for evaluating the performance of human detection systems are consist of still images and the main idea of this thesis is to use motion information for improving the detection results we had to adapt datasets from other domains of machine vision to test our method. We used two different publicly available datasets to evaluate our method and we ran two sets of different experiments on these datasets. The first dataset is the ETHZ crossing dataset. This dataset is used in Leibe et al. [19] for the coupled detection and tracking. The second dataset is the VIRAT video dataset, this dataset is introduced in Oh et al. [20], the main purpose of this dataset is to provide a benchmark for event recognition in surveillance video. We used a subset of this dataset to test our method.

4.1 ETHZ Central Pedestrian Crossing

This dataset is introduced for the tracking task and since most of the tracking methods heavily relies on the detection results it seems to be a good benchmark for our pedestrian detection method. This dataset consists of 3 sequences which are recorded using a public webcam at 15fps, 320 $\times$ 240 pixels resolution and they contain severe MPEG compression artifacts. In the scenes of this dataset we have different moving objects, cars, motorbikes and trains as well as the objects which are carried by the pedestrians such as suitcases, strollers and shopping carts. This variety of objects and different types of occlusions available in this dataset makes it a good choice for testing our method.

Each frame of this dataset contains at least 2 pedestrians, this dataset contains 1400 frames and in total it does have 5000 pedestrians. We spotted a few issues when we used this dataset. First of all there are some inconsistencies in the labelling where some pedestrians are not labelled in some frames while they are labelled in other frames. The other issue with this dataset is that when the pedestrians go under trees they are not labelled. However, since one of the advantages of our method is to detect partially occluded pedestrians we relabelled the dataset and labelled those pedestrians.

4.2 Experiments

We ran 3 sets of experiments on this dataset with 3 different setups. We compared the results of our method with the results of our baselines, HOG and HOF. In all of the experiments we did we used 300 frames of the first sequence to do the training. This dataset provides the bounding box for the pedestrians which we used as positive examples but for the negative example we have to generate them ourselves. To do so we used a window of a random size, the mean of this random size is the mean size of bounding boxes in the dataset and we set a high and low value for each bounding box and uniformly generated bounding boxes and then randomly put the bounding box on the image, if the bounding box has less than 50% overlap with positive examples of the frame we accept it as a valid negative example. For the first round of the training we generated 20 negative examples per frame which means we would have a total of 6000 of negatives while having 650 positive samples.

We trained our latent model using the positive and negative examples we generated using the above mentioned algorithm. Then we used the obtained classifier and used it to detect positive examples in the training dataset using a sliding window scheme where the size of the window is the average size of the bounding boxes of the dataset, then we used the top ten false positives in each frame and added them to the training set to do a round of bootstrapping. We trained our classifier again to achieve the final classifier. We do the same procedure for HOG and HOF, we use the same initial training set for all the methods but for the bootstrapping step we extract the hard negatives for each method separately.

We ran 3 sets of experiments on this dataset. In the first test we used the dataset in its original format without any changes to the dataset, as we can see in Fig. 4.1 our methods performs slightly better than the other method especially at the false positive rate of 10^{-4}, which in our resolution means 1 false detection per image, our miss rate is lower.

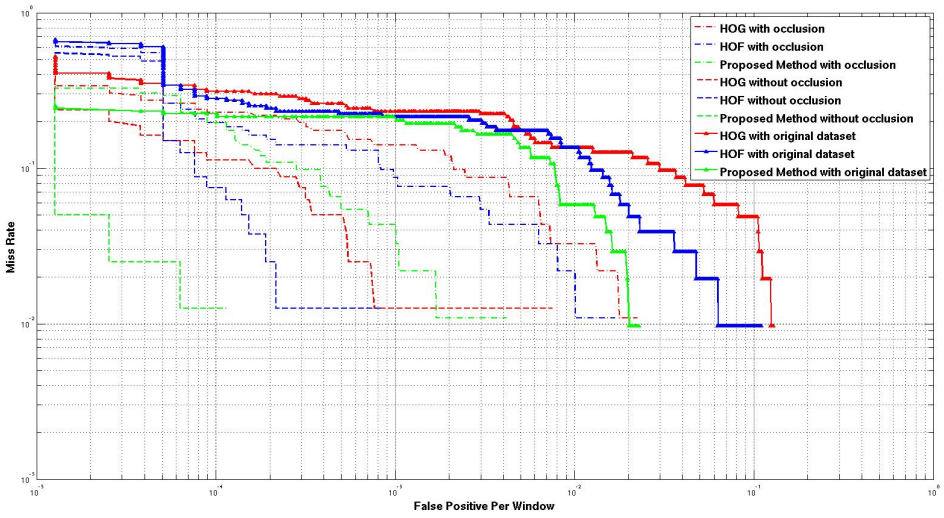

Figure 4.1: Comparison of different methods results on ETHZ crossing dataset.

Some of the false positives for our method comes from the point that we do the detection in the occluded area and those areas are marked as negative in the annotation. So we relabelled the dataset by putting bounding boxes over occluded pedestrians as well as completing the labelling for the misclassified pedestrians and relabelled the dataset. With this new relabelled dataset we extract the negative examples again and train our classifier again. As it can be seen in Fig. 4.1, the performance of our system gets much better than the other two systems specially in lower miss rates.

The last test we did on this dataset shows the performance of our system where there is no major occlusion. To that end we eliminated the occluded pedestrians as well as some pedestrians in the corners of the image are way too small and highly occluded with fences (these pedestrians are mainly the ones which were mislabelled in the original dataset) and then with this newly labelled dataset we retrain everything and then test the results of our algorithm on this dataset. Since this dataset has a much easier setup all the methods produce good results on the test set. But our method is significantly better than the 2 other methods. Some example detection results of our method on ETHZ dataset can be seen in Fig. 4.2.

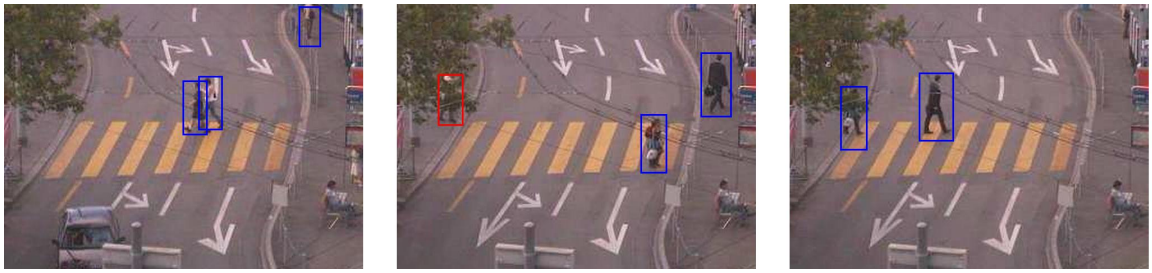

Figure 4.2: Detections on ETHZ dataset(blue boxes are true positive, red box is the false negative)

4.3 VIRAT Video Dataset

This dataset is a benchmark for human event detection, especially single person events, person and vehicle events and person facility events. This dataset consists of many natural outdoor scenes with actions occurring by non-actors in continuously captured videos from security cameras. The dataset includes large numbers of instances for 23 event types distributed throughout 29 hours of video. This data is accompanied by detailed annotations which include both moving object tracks and event examples. Since the data for this dataset is captured from various cameras and locations, there is a great variety in the resolution of the videos as well as scenes and the scales of the people in the videos.

The number of pedestrians in each frame varies between different sequences ranging from no pedestrians for a long time to around 10 pedestrians per frame. Since this dataset is very huge and has a lot of variety in it we initially took 12 sequences which have different types of scenes and setups. Since we introduced a detection method we try to evaluate its performance only and avoid the other parameters which can affect the performance of the detection system, so we used the sequences with limited perspective changes because if there is a great amount of the perspective changes in the scene we have to use camera calibration and rescale each part of the image separately and do a lot of engineering to make the systems work so we try to avoid this part by choosing the input sequences wisely.

Although this dataset is widely used for the human event detection, unfortunately there are so many mislabelled subjects in its ground truth. Further some sequences missed humans in the whole sequence. Because of this we had to remove 2 out of the 12 videos from our dataset even though we got good qualitative results from those sequences. However, since we couldn't provide any quantitative results for those sequences we had to remove them

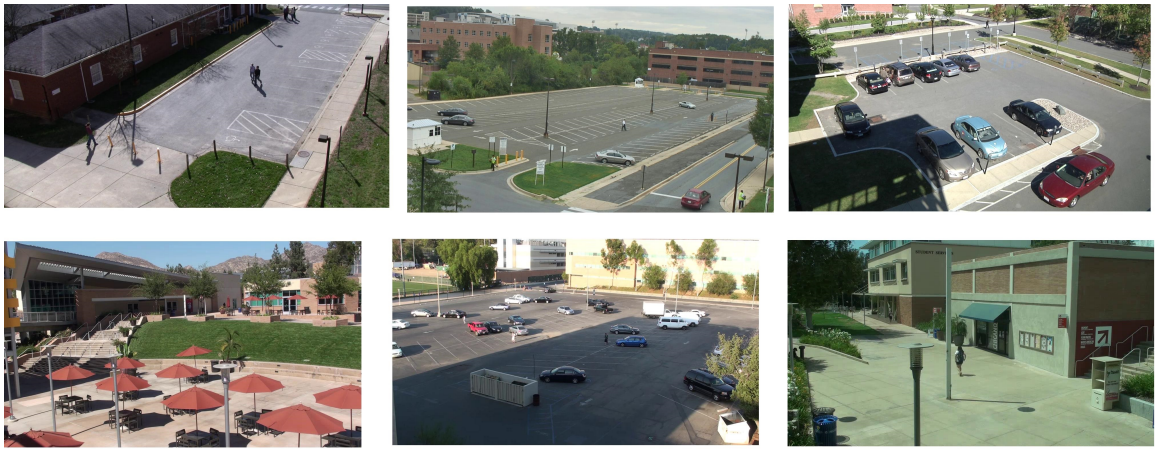

Figure 4.3: Different scenes of the VIRAT dataset.

from the results.

4.4 Domain Adaptation

In the first round of experiments with the ETHZ dataset we showed that our method outperforms the other 2 baselines when all the methods are trained on the parts of the dataset which they are going to be tested on. This is very common in testing most of the detection systems. But a more difficult and more realistic task is to train the detectors on some dataset and then test the classifier on an unseen dataset which is called domain adaptation. This task is very useful because if it can be done successfully, we can save the time and money for labelling a new dataset and also it makes the detection system more automatic. We tried to do this task on this dataset so we trained our baselines on the ETHZ crossing dataset then we tested our method on the VIRAT dataset. Unfortunately this experiment failed and none of the classifiers could produce acceptable results on this task. So we tried to improve the performance of the system by adding some realistic assumptions to the system. Although labelling the positive examples in different sequences is a very time consuming and expensive task but we can extract some reliable negative examples. The idea behind getting the negative samples is simple, since we know that usually most of the humans in the sequences walk so they don't stay at one place for too long. Using this fact, we extract the background using median filtering. Since humans are usually walking it is very unlikely that this background contains any human. Having the background as our

negative we train our systems on the ETHZ crossing dataset and then test these models on the background and use the results of this experiment and do a round of bootstrapping. This round of bootstrapping is significantly improving the performance our detectors. The comparison between results of our method and the other two baselines can be seen in Fig. 4.4.

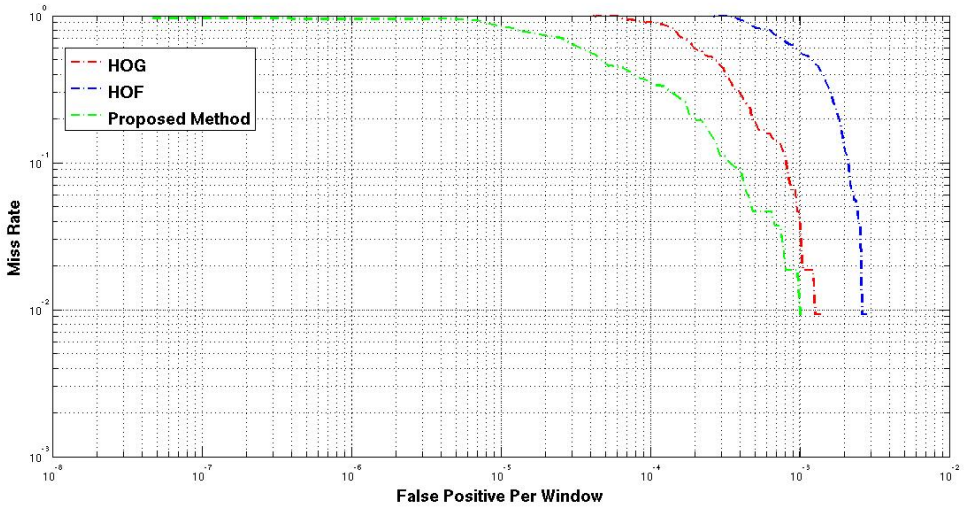

Figure 4.4: Comparison of different methods results on VIRAT dataset.

The basic idea behind our domain adaptation system is simple but we have to consider many minor and yet important points when trying this algorithm. The first issue is the frame rate which may cause difficulties when doing the domain adaptation. The ETHZ dataset is captured at the rate of 15 fps which is uncommon but it is very useful because we can use 5 frames for the tracking and get enough information for detecting the motion patterns but when we are using another test dataset with different fps we should consider this difference. For compensating the difference between fps of two sequences we downsampled the VIRAT dataset and since VIRAT is captured at 30 fps we only need to skip every other frame to reach the correct performance.

Another practical point about the VIRAT dataset is that this dataset is captured in higher quality than what we had in the ETHZ dataset. Although we can compute the SOEs on the original format of the dataset but that would be very computationally expensive and also it need a lot of space to store the SOEs. So we resized the original dataset to reduce the computation overhead but since we can run HOG and HOF in the original size

of the dataset it is not fair for them to downsample the dataset for them so we produce the results for these algorithm in the original size of the dataset. This difference between working resolution may lead to some difficulty when we want to compare the results of our method to the other methods. We can use two different approaches to solve this problem and they produce almost same results. In the first method we can downsample the results of the other methods to the resolution of our method and then compare the results. Although this is a valid comparison, it might seems unfair to the methods which can work in high resolution. So in the second solution we resized the results of our method to the results of other methods using bilinear interpolation. Both methods show similar results and in both comparisons proposed method outperforms the baselines.

4.5 Implementation Details

There are a few tricks and tweaks in the system to get reasonable results, here we discuss those tweaks. First of all we are using SOEs with 12 channels. An observation is that the SOE values are consistent within a video but not across the videos if we don't do any normalization. So we use the 12 channel energies and add a channel which captures the energy of unstructured pixels and normalize the SOEs using that extra channel.

Another important point for our current detection system is since we don't attempt to detect pedestrians walking straight to the camera and all the pedestrians we are trying to detect are moving with some kind of angle to the camera and unlike Felzenszwalb's [15] paper we don't have a latent value for the direction of pedestrian so we use the SOE channels to estimate the direction of the movement and normalize the tracklet so the subjects move from left to right.

We are proposing a detector here and we don't attempt to improve the results with NMS or any other kind of post processing system so in the generation of the final Detection Error Tradeoff (DET) curves we use Dalal and Triggs method which means we only need one detection with over 50% overlap with the ground truth to mark the subject as a hit.

We should also mention that the current detection system ignores the pedestrians who are too close to image boundaries and leave the scene during the tracking which reduce the performance of our detector because the HOG and HOF still have a chance to detect these pedestrians if they are visible in the first frame.

Some of the interesting results of our detector is that it can detect partially occluded

pedestrians based on their motion patterns, we can see examples of these detections when the subjects goes under tress or behind patio umbrellas. Another interesting point is although we scale down our input image to handle the computation complexity of the SOE and tracking we still can detect subjects with small bounding boxes which is very important because most of the current detectors can't perform well enough in low resolution or when the subject is small, but it seems that the motion patterns carry enough information to improve the detection. The detection results of our method on VIRAT dataset can be seen in Fig. 4.5.

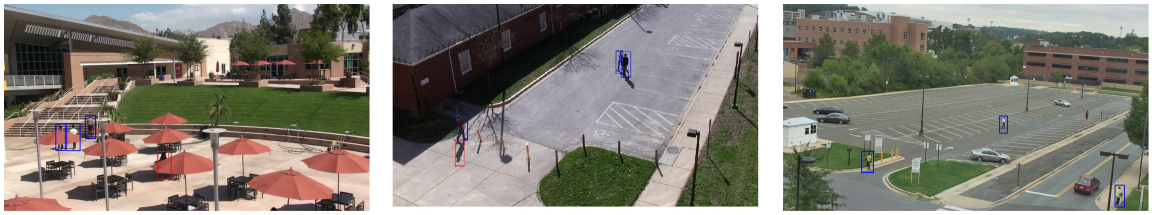

Figure 4.5: Detections on the VIRAT dataset.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

The main contribution of this thesis is developing a human detection system using tracklets of different body parts as motion pattern. We proposed a framework which can consider the motion pattern as well as the appearance of the subject. The previously introduced models only rely on the appearance or only two consequent frames for the motion model which doesn't really capture the rhythmic motion of the human motion.

The most important draw back of the proposed method is that it wouldn't improve detection over the static pedestrians and although this method can detect partially occluded pedestrians it can mainly handle the occlusions in upper parts of the body and if the occlusion happens over the lower parts of the body the improvements wouldn't be significant.

5.2 Future Works

There are some drawbacks in the current system which can be fixed in future works. The first important issue is that the performance of the system would increase if we could align the sequences based on the gait phase of the pedestrian. This alignment can be done using a latent variable representing the phase of the gait, so we can compare the aligned sequences which would increase the accuracy of our detector.

The other modification which can improve the results of our method is that if we can detect the direction of the movement and use the SOEs of that direction and apply a HOG like operator on the SOE we will have a detector which works similar to HOF and it might

be able to do the cross learning. This extra detector can help and improve the performance of the system.

We evaluated the performance of our system without considering the rule of NMS but a good NMS system can improve the performance of the system so in the future work we should investigate the improvements of the different NMS schemes. To have a fully working system we have to add the NMS and optimize it parameters.

In this thesis we didn't consider the different scaling issue but in real world many scenes have different scaling factors in a scene and we have to find a way to handle it. This issue can be resolved using the calibration information of the camera or the scene configuration.

A good improvement of our system would be to replace the HOG detector by the model proposed in [15] and do the detection. This addition would improve our results since this method is a better detector than the HOG detector. Another possible way to continue our work is to detect the parts using Felzenszwalb's method [15] and then track those parts. This way we might have better parts to track so the tracklets would be more meaningful. This improvement would be mainly because this way we would actually track the different parts of the body and these tracklets would have more meaning than just tracking different parts of the bounding box.

Another possible improvement for the current system would be to train different classifiers for different view points, as we already mentioned most of our training and testing happens in side views but we usually won't get good results if the subjects move straight forward to the camera. It would be possible to use exemplar SVM or clustering to determine the type of the subject and use the proper classifier to detect class of the subject.

Bibliography

[1] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *JOSA-A*, 2(2):284–299, 1985.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR08*, pages 1–8, 2008.

[3] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, February 1994.

[4] M. Brand and V.M. Kettnaker. Discovery and segmentation of activities in video. *PAMI*, 22(8):844–851, August 2000.

[5] K. Cannons and R. Wildes. Spatiotemporal oriented energy features for visual tracking. In *ACCV 2007*, pages 532–543. 2007.

[6] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] R.T. Collins. Mean-shift blob tracking through scale space. In *CVPR03*, pages II: 234–240, 2003.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR05*, pages I: 886–893, 2005.

[9] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV06*, pages II: 428–441, 2006.

[10] K.G. Derpanis and J.M. Gryn. Three-dimensional nth derivative of gaussian separable steerable filters. In *ICIP05*, pages III: 553–556, 2005.

[11] Trinh-Minh-Tri Do and Thierry Artières. Large margin training for hidden markov models with partially observed states. In *ICML09*, pages 265–272. ACM, 2009.

[12] P. Dollar, S.J. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC10*, pages xx–yy, 2010.

[13] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34(4):743–761, April 2012.

[14] M. Enzweiler and D.M. Gavrila. Monocular pedestrian detection: Survey and experiments. *PAMI*, 31(12):2179–2195, December 2009.

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

[16] A. Garcia Martin, A. Hauptmann, and J.M. Martinez. People detection based on appearance and motion models. In *AVSBS11*, pages 256–260, 2011.

[17] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Who? when? where? what? a real time system for detecting and tracking people. In *AFGR98*, pages 222–227, 1998.

[18] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. pages 145–153, 2004.

[19] B. Leibe, K. Schindler, and L.J. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV07*, pages 1–8, 2007.

[20] S.M. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.C. Chen, J.T. Lee, S. Mukherjee, J.K. Aggarwal, H. Lee, L.S. Davis, E. Swears, X.Y. Wang, Q.A. Ji, K.K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy Chowdhury, and M. Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR11*, pages 3153–3160, 2011.

[21] C.P. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38(1):15–33, June 2000.

[22] T. Serre, L. Wolf, S.M. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *PAMI*, 29(3):411–426, March 2007.

[23] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR12*, pages 1815–1821, 2012.

[24] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, July 2005.

[25] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *CVPR10*, pages 1030–1037, 2010.

[26] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV05*, pages I: 90–97, 2005.

[27] J. Yan, Z. Lei, D. Yi, and S.Z. Li. Multi-pedestrian detection in crowded scenes: A global view. In *CVPR12*, pages 3124–3129, 2012.

[28] Q.A. Zhu, M.C. Yeh, K.T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR06*, pages II: 1491–1498, 2006.