Techniques for Detection and Tracking of Multiple Objects

Mohamed Naiel

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québéc, Canada

January, 2017

Concordia University

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Mohamed Naiel**

Entitled: **Techniques for Detection and Tracking of Multiple Objects**

and submitted in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY (Electrical and Computer Engineering)** complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. S. V. Hoa	Chair
Dr. P. Agathoklis	External Examiner
Dr. C.-Y. Su	External to Program
Dr. W.E. Lynch	Examiner
Dr. W.-P. Zhu	Examiner
Dr. M.O. Ahmad	Thesis Co-Supervisor
Dr. M.N.S. Swamy	Thesis Co-Supervisor

Approved by _____
Dr. W.-P. Zhu, Graduate Program Director

January 30, 2017.

_____
Dr. A. Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Techniques for Detection and Tracking of Multiple Objects

Mohamed Naiel, Ph.D.

Concordia University, 2017

During the past decade, object detection and object tracking in videos have received a great deal of attention from the research community in view of their many applications, such as human activity recognition, human computer interaction, crowd scene analysis, video surveillance, sports video analysis, autonomous vehicle navigation, driver assistance systems, and traffic management. Object detection and object tracking face a number of challenges such as variation in scale, appearance, view of the objects, as well as occlusion, and changes in illumination and environmental conditions. Object tracking has some other challenges such as similar appearance among multiple targets and long-term occlusion, which may cause failure in tracking. Detection-based tracking techniques use an object detector for guiding the tracking process. However, existing object detectors usually suffer from detection errors, which may mislead the trackers, if used for tracking. Thus, improving the performance of the existing detection schemes will consequently enhance the performance of detection-based trackers. The objective of this research is two fold: (a) to investigate the use of 2D discrete Fourier and cosine transforms for vehicle detection, and (b) to develop a detection-based online multi-object tracking technique.

The first part of the thesis deals with the use of 2D discrete Fourier and cosine transforms for vehicle detection. For this purpose, we introduce the transform-domain two-dimensional histogram of oriented gradients (TD2DHOG) features, as a truncated version of 2DHOG in the 2DDFT or 2DDCT domain. It is shown that these TD2DHOG features obtained from an image at the original resolution and a downsampled version from the same image are approximately the same within a multiplicative factor. This property is then utilized in developing a scheme for the de-

tection of vehicles of various resolutions using a single classifier rather than multiple resolution-specific classifiers. Extensive experiments are conducted, which show that the use of the single classifier in the proposed detection scheme reduces drastically the training and storage cost over the use of a classifier pyramid, yet providing a detection accuracy similar to that obtained using TD2DHOG features with a classifier pyramid. Furthermore, the proposed method provides a detection accuracy that is similar or even better than that provided by the state-of-the-art techniques.

In the second part of the thesis, a robust collaborative model, which enhances the interaction between a pre-trained object detector and a number of particle filter-based single-object online trackers, is proposed. The proposed scheme is based on associating a detection with a tracker for each frame. For each tracker, a motion model that incorporates the associated detections with the object dynamics, and a likelihood function that provides different weights for the propagated particles and the newly created ones from the associated detections are introduced, with a view to reduce the effect of detection errors on the tracking process. Finally, a new image sample selection scheme is introduced in order to update the appearance model of a given tracker. Experimental results show the effectiveness of the proposed scheme in enhancing the multi-object tracking performance.

Acknowledgments

I would like to express my deepest gratitude and appreciation to my supervisors, Prof. M. Omair Ahmad and Prof. M.N.S. Swamy, for their constant support, encouragement, patience, and invaluable guidance at every phase of this research. I would also like to express my deep appreciation for their encouragement to participate with them in two industrial projects under their supervision and leadership. I also extend my thanks and appreciation to the committee members for the useful suggestions.

I am very grateful to my supervisors and Concordia University for the financial support that I received, which was very crucial for completing this research work. I would like to acknowledge the financial support from the School of Graduate Studies, Faculty of Engineering and Computer Science, GSA and ECSGA at Concordia University, NSERC, ReSMiQ and the supervisors for covering the publication costs related to this research. I acknowledge the support and help offered to me by the Department of Electrical and Computer Engineering, Concordia University. Further, I would like to acknowledge the support of Mitacs and MakerBloks, as well as the Ville de Montreal during my work with Concordia University in two industrial projects.

My special thanks to all my fellow colleagues at the Center for Signal Processing and Communications, Concordia University, specially, Hamidreza Sadreazami, Marzieh Amini, Amgad Salama, Shreyamsha Kumar, Masoumeh Abkenar, Waziha Kabir, Prashanth Venkataswamy, Omid Saatlou, and Homa Tahvilian. My profound gratitude goes to my parents, brothers and sisters for their continues encouragement during my doctoral study. I would specially like to thank my father, Ahmed Naiel, for his continues support and encouragement to excel in my studies, life and career. I dedicate this research work to the soul of my dear mother, Sabah, who passed away after the defence of my thesis. Special thanks also to my brother, Ehab Naiel, for his partial financial support during my doctoral degree program. Last but not least, I would like to express my deep love and appreciation for my lovely daughter, Layla Naiel, whose presence has given me strength and peace of mind.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbf{A}$	Matrix consists of column vector versions from the training samples, and of size $(r \times N^t)$, where $r = m_1 m_2$
$\mathscr{B}^t$	Set of trackers at time t
b	Block size to compute the image transform (i.e., 2DDFT or 2DDCT)
b_0	The block size used at $R = 1$, referred to as *base block size*
b_t	Tracker identity at time t
$\mathbf{Cov}$	Covariance matrix
c, c_1, c_2	The maximum frequencies retained by the truncation operator
$\mathbf{D}$	Dictionary of N_k local patches, where $\mathbf{D} \in \mathbb{R}^{\hat{r} \times N_k}$ $(\hat{r} = \hat{m}_1 \hat{m}_2)$
$\mathscr{D}^t$	Set of detections at time t
d_t	Detection identity at time t
$\mathbf{F}$	The transition matrix of the motion model
$\mathcal{F}_R$	The set of feature vectors obtained from $\mathcal{I}_R$
$\mathbf{f}$	TD2DHOG feature vector after 2DPCA projection
G_{PGM}	The similarity measure of PGM
G_{SGM}	The similarity measure of SGM
$G[u, v]$	The $(u, v)^{th}$ element of two dimensional discrete signal coefficients in the transform domain (i.e., 2DDCT/2DDFT)

$\hat{G}[u,v]$	The $(u,v)^{th}$ element of downsampled 2D discrete signal coefficients in the transform domain (i.e., 2DDCT/2DDFT)
$\mathcal{G}_{b_t}$	A gate function that represents the state of the tracker b_t when associated to the detection d_t at time t
$g_x(i,j), g_y(i,j)$	Gradients in the x and y directions, respectively, where i and j denote the pixel indices
H_{SDC}	SDC tracker confidence score
$\mathbf{H}^l$	The features of the l^{th} layer in the 2DDFT or 2DDCT domain and of size $(N_1 \times M_1)$
$\hat{\mathbf{H}}$	TD2DHOG features
$\bar{\mathbf{H}}$	The average over N_t TD2DHOG features
$\mathbf{h}, \mathbf{H}$	2D image features in the spatial or the transform domain, respectively, each of size $(N_1 \times M_1 \times \beta)$, such as 2DHOG features
$\mathbf{I}$	An input image or detection window
$\mathbf{I}_s$	Image resampled by a factor s
$\mathcal{I}_R$	A set of training data at which all training images are upsampled by a factor R
$\mathbf{J}(.)$	The total scatter criterion of PGM
K, K_1, K_2	Downsampling factors
K_u^t	Selected key samples fat time t
k, u, v	The transform domain discrete sample index
$L, L^\top$	High pass filter in the x direction and its transpose in the y direction
l	2DHOG layer number
M_1, M_2	The size of an image or detection window in pixels
$\tilde{M}_1, \tilde{M}_2$	The size of each 2DHOG layer in the spatial domain
$\hat{M}_1, \hat{M}_2$	The size of each TD2DHOG layer $\hat{H}^l$

m_1, m_2	The size of each sample image in pixels drawn to construct the appearance model of a tracker
N_p, N_n	Number of positive and negative samples drawn to initialize a tracker
$N^t_{p,u}, N^t_{n,u}$	Number of positive and negative samples drawn at time t to update a tracker
N_c, N_{c_1}, N_{c_2}	Low pass filter cutoff frequencies
N_{pos}, N_{neg}	The total number of positive and negative training image samples in a dataset
N_x, N_y	Number of 2DDCT or 2DDFT blocks in x, and y directions, respectively
N_t	Number of training examples
n, m	The spatial domain discrete sample index
$O(b_t, d_t)$	The overlap ratio between the tracker, b_t, and the detection, d_t
O_1	Overlap threshold
$\mathcal{P}(\mathbf{I}, s)$	Resampling operator in the spatial domain to resample the image $\mathbf{I}$ by a factor s
$\mathbf{Q}^l$	Matrix constructed after projecting the TD2DHOG of the l^{th} layer on the matrix $\mathbf{V}^l$
$\mathbf{q}^l$	TD2DHOG feature vector of the l^{th} layer
$q(.)$	The importance density function of the particle filter
R	Upsampling factor
R_u	Updated rate of a tracker
$\mathbb{R}$	Real space
r	The size of the feature vector $\mathbf{f}$
r_l	The number of eigenvectors retained for the l^{th} TD2DHOG layer

Re(.)	A function which returns the real part of an input complex number
S	Similarity matrix used to solve the data association problem
s	Image resampling factor, where $s < 1$ represents downsampling, $s > 1$ represents upsampling
s_0, s_1, s_2	Similarity thresholds
$T(.)$	Represents the image transform
$\hat{T}(.)$	Represents the transform operation followed by the truncation operation
$\mathcal{T}_R$	A classifier trained at the upsampling factor R by using the corresponding features $\mathcal{F}_R$
t	Time
$\mathbf{V}^l$	Matrix of size $(\hat{M}_2 \times r_l)$ that includes the r_l eigenvectors correspond to the dominant eigenvalues
$\mathbf{V}_{opt}$	The optimal orthonormal matrix of eigenvectors
$\mathcal{V}_R$	The set of eigenvectors obtained from $\mathcal{I}_R$
$X[k], Y[k], Z[k]$	One dimensional discrete signal in the frequency domain
X^{b_t}, X^{b_t,d_t}	The set of propagated particles and associated detections, respectively
$x[n], y[n], z[n]$	One dimensional discrete signal in the time domain
$\mathbf{x}_Q, \mathbf{x}_P$	Gaussian noise vectors
$\{\mathbf{x}_t^i, \boldsymbol{w}_t^i\}_{i=1}^{N_s}$	The set of particle state vectors and its the corresponding weights, where $\boldsymbol{w}_t^i$ is the weight of particle, $\mathbf{x}_t^i$, and N_s is the total number of particles
$\mathbf{x}$	The state vector that consists of translation (x, y), average velocity (v_x, v_y), scale $\hat{s}$, rotation angle θ, aspect ratio η, and skew direction ϕ

$\bar{\mathbf{Y}}$	The average image of all training image samples
$\{\mathbf{y}_i\}_{i=1}^{M}$	Set of M local patches each patch of size $\hat{m}_1 \times \hat{m}_2$ for SGM tracker
$\mathbf{z}, \mathbf{z}_s$	The exact 2D features extracted from an image at the original resolution, and the same image at a different resolution, s, respectively.
$\tilde{\mathbf{z}}_s$	The approximated 2D features for the features obtained from an image resampled by a factor s
$\mathbf{z}_t$	The measurement vector of the particle filter at time t
α	Multiplicative factor for the transform domain
$\hat{\alpha}$	The estimated multiplicative factor from N_t images
$\tilde{\alpha}$	Sparse coefficients
β	Number of 2DHOG or TD2DHOG layers
$\tilde{\beta}_i$	The sparse-coefficients, $\tilde{\beta}_i \in \mathbb{R}^{N_k \times 1}$, of the i^{th} patch, y_i, for the SGM tracker
$\Gamma(i,j), \theta(i,j)$	The magnitude, and orientation of the gradient at (i,j), respectively
$\hat{\Gamma}_N$	Weighting term of 1DDCT for a sequence of length N
γ_{CF}	Tracker collaborative factor
γ	Multiplicative factor for approximation in the spatial domain
$\delta_l(i,j)$	Function that returns 1 if $\hat{\theta}(i,j) = \Omega(l)$ and 0 otherwise
$\delta_j^{'}$	A function that selects the coefficients corresponding to the j^{th} class and suppresses the rest
$\{\varepsilon_i\}_{i=1}^{M}$	The patch reconstruction error of SGM, which is used to suppress the coefficients of occluded patches
ε_{PGM}	The reconstruction error between the test image and the training examples

$\varepsilon_+^i, \varepsilon_-^i$	The reconstruction error of the candidate $\mathbf{z}_t^i$ with respect to the template set of the positive or negative class, respectively
η_1, η_2	2DHOG cell size in pixels
$\hat{\theta}(i,j), \Omega(.)$	The quantized orientation at $(i,j)^{th}$ pixel and the range of quantized β angles for obtaining 2DHOG layers, respectively
$\Lambda(.)$	A 2D spatial-domain feature extractor
λ, a_0, a_1	Channel parameters for TD2DHOG features
$\lambda_{SDC}, \lambda_{SGM}$	Weighting factors for the regularization terms that exist in the optimization problems of SDC and SGM, respectively
μ	The learning rate of SGM tracker
ρ, ρ^c	The sparse histogram feature vector of the object and the candidate location, respectively.
σ	Parameter used to adjust the confidence score of the SDC tracker
$\boldsymbol{\phi}_{c_1 c_2}(.)$	2D truncation operator in the transform domain that truncates the coefficients corresponding to the frequencies greater than the frequencies c_1 and c_2
φ, φ^c	The sparse histograms after employing occlusion handling scheme of SGM, which correspond to the training template, and the candidate location, respectively
ψ_i	Non-occlusion indicator for the i^{th} patch
$\Omega_{SCI}(.)$	The sparsity concentration index

List of Abbreviations

2DDCT	Two-Dimensional Discrete Cosine Transform
2DDFT	Two-Dimensional Discrete Fourier Transform
2DHOG	Two-Dimensional HOG
2DPCA	Two-Dimensional Principal Component Analysis
3DVP	3D voxel patterns
AdaBoost	Adaptive Boosting
ACF	Aggregated Channel Features
AP	Average Precision
AFP/F	Average False Positive per Frame
AFP/O	Average False Positive per Object
ATP/F	Average True Positive per Frame
BDTC	Boosted Decision Tree Classifier
CP	Classifier Pyramid
CTT	Classifier Training Time in Second
DT	Detection Time in Second
DPM	Deformable Part-based Model
EER	Equal Error Rate
FDR	False Detection Rate
FET	Feature Extraction Time in Second
FFT	Fast Fourier Transform
FPS	Frames Per Second

FPD	Fast Pedestrian Detector
FNR	False Negative Rate
FPR	False Positive Rate
FIKSVM	Fast Histogram Intersection Kernel SVM
GS	Grayscale
GT	Ground Truth
HOG	Histogram of Oriented Gradients
H-dense	Highway of high density
H-medium	Highway of medium density
IDSW	Identity Switches
IN	Initialization
ISM	Implicit Shape Model
LSVM	Latent Support Vector Machine
MAE	Mean Absolute Error
MOT	Multi-Object Tracking
MSE	Mean Square Error
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
neg	Negative
OH	On-hold
PCA	Principal Component Analysis
PGM	2DPCA-based Generative Model
pos	Positive
SC	Single Classifier
SVM	Support Vector Machine
SGM	Sparsity-based Generative Model
SDC	Sparsity-based Discriminative Classifier
SCI	Sparsity Concentration Index

TD2DHOG	Transform Domain Two-Dimensional Histogram of Oriented Gradients
TPR	True Positive Rate
TT	Average Training Time in Seconds
TRM	Termination
UCF-PL	University of Central Florida Parking Lot Dataset
VOC	Visual Object Classes

Chapter 1

Introduction

1.1 General

The past decade has witnessed significant progress in the computational power and storage capability of portable computers, reliability and speed of several types of communication networks, and high-resolution digital cameras, as well as a huge increase in the number of Internet and social media users. These developments have resulted in an explosive growth in visual information that have introduced new challenging problems in the computer vision field to automatically analyze and understand this information. In this context, object detection and tracking in videos can be the two main building blocks for several computer vision applications, such as human activity recognition, human computer interaction, crowd scene analysis, video surveillance, sports video analysis, autonomous vehicle navigation, driver assistance systems, and traffic management.

In computer vision literature, various techniques have been introduced in order to tackle the problem of detection of pedestrians and vehicles in images and videos. There has been a great deal of work carried out in this field [1–4], especially, in designing an object detector that takes into consideration the changes in scale, appearance, view of the objects, as well as partial occlusion, and changes in the il-

lumination conditions. Multi-object tracking (MOT) is another challenging problem in computer vision, which has numerous applications such as automatic visual surveillance, behavior analysis, and intelligent transportation systems. Recently, detection-based tracking has received considerable attention [5]. In detection-based tracking, an object detector that has been trained on a specific class (for example, cars) is used to guide multiple object trackers. The main objective of multi-object tracking techniques is to maintain the identity of the objects through a given video sequence by solving an association problem among detections and trackers. Existing object detectors usually suffer from false positive and missed detections, which may misguide the tracker, when used for tracking. Thus, the design of an object detector that offers a high detection accuracy is a pre-condition for obtaining a detection-based tracker that is able to follow changes in the object appearance through time.

1.2 Literature Review

In this section, a review on some of the recent advances in detection and tracking of multiple objects is presented.

1.2.1 Object Detection

For the purpose of object detection and recognition, several types of image features and their representations, such as the histogram of oriented gradients (HOG) [6], Haar-like features [7], interest-points based features [8–12], shape context [13], local binary patterns [14], and 3D voxel patterns (3DVP) [15], have been introduced. The various schemes for object detection may be categorized into three main types, namely, *sliding window-based methods*, *part-based methods*, and *interest point-based methods*.

In the *sliding window-based methods*, features of a certain type are obtained for the entire object. For instance, in 1998 the Haar-wavelet basis functions were introduced

by Papageorgiou *et al.* [7] for face or pedestrian detection. The Haar-wavelet was used to extract an overcomplete set of features from the target object, followed by a feature selection technique, where the support vector machine (SVM) classifier was used. Later, Viola and Jones [16] introduced the Haar-like features for fast computation of an approximated version from Haar-wavelet and used the adaptive boosting (AdaBoost) technique for feature selection. These Haar-like features have been employed widely in several application domains, such as face detection [17], pedestrian detection [18], and object tracking [19].

In 2005, Dalal and Triggs [6] introduced a human detection algorithm that is based on HOG features with a linear SVM for classification. Later, HOG features have been investigated widely and used in the state-of-the-art techniques for object detection and description [4]. Instead of the 1D vector representation of HOG [6, 20, 21], several papers have adopted a 2D representation [22–24], since the latter preserves the relations among the neighboring pixels or cells. In order to distinguish the 2D representation from the 1D one, we will call it 2DHOG. Both the 1D and 2D representations of HOG capture the edge structure of the object and are robust against illumination changes and background clutters. However, neither of these representations is resolution invariant. Thus, detectors employing these representations require extracting HOG or 2DHOG features at each scale from an image pyramid, thus requiring a costly multi-scale scanning in the testing mode [22, 23].

There are several works that have been introduced to reduce the complexity of computing the HOG features [23, 25–28]. For instance, Zhu *et al.* [25] introduced a computationally fast method for obtaining HOG features using integral histograms [29]. In this method, AdaBoost [30] was used with HOG as a feature selection technique from a large set of features, where the SVM classifier was used as a weak classifier for every consistent set. However, AdaBoost requires expensive parameter tuning, and thus, a high training cost. Dollár *et al.* [23] combined multiple feature channels, such as grayscale, gradient magnitude and 2DHOG, with a modified

AdaBoost based on using the multiple-instance pruning algorithm [31] to overcome the expensive parameter tuning of the original one [30]. Even though the integral images and integral histograms were used for the fast computation of the channel features, this method needs the extraction of features from an image pyramid, thus resulting in a high computational cost in the testing mode.

Recently, Dollár *et al.* [32, 33] proposed a feature approximation technique, where gradient histograms and color feature responses generated at one scale of an image pyramid can be used to approximate the feature responses at nearby scales. This method results in a speedup of extracting the features from the image pyramid over the methods of [22, 23], with only a small reduction in the detection accuracy. In this technique, the feature responses can be approximated with high accuracy within one octave of the scales of the image pyramid. Later, authors in [34, 35] enhanced the detection performance of [32] by constructing a classifier pyramid instead of an image pyramid. However, since the methods in [34] and [35] are based on constructing a classifier pyramid with multiple classifiers trained at different sizes of the object, they require a high training and storage cost.

The *part-based methods* have received a great deal of attention from the research community, as these schemes can handle partial occlusion, and represent targets with several views [18, 24, 36–42]. The general idea in part-based techniques is that an object of interest is divided into a number of components and every component is detected separately, where a fusion rule is used for combining the results of multiple detected components. For instance, Mohan *et al.* [18] considered the human body to consist of four components (face, legs, and left and right arms) and proposed a classifier for each of the components. In this method, a two-level detector was used. In the first level, the Haar-wavelet was obtained, and for each body part a SVM classifier was trained. Then in the second level, the responses of the four detectors were fused using another SVM classifier to obtain the final decision.

In 2005, Felzenszwalb and Huttenlocher [38] introduced a learning technique for

generic part-based models. Inspired by the pictorial structure representation in [43], each local part of the object is assumed to encode the local part visual properties, and a deformable configuration is used to represent the relationship between the local parts. Later, Felzenszwalb *et al.* [24] have proposed a pictorial structure for HOG features, referred to as deformable part-based model (DPM). In this method, the locations of the parts are used as latent variables for a latent support vector machine (LSVM) classifier to find the optimal object position. Later, several other techniques adopted DPM [24] for vehicle detection [41, 44, 45], providing a high detection accuracy. However, they require convolutions of the features of a given level of the image pyramid with a number of part filters, resulting in a high computational cost.

Some of the latest schemes in the area of object detection [15, 46, 47] have attempted to overcome the problems concerning scale, aspect ratio or severe occlusion. For example, the method in [46] has used a detection scheme based on the DPM detector [24] and introduced a method for clustering the training data into a number of similar occlusion patterns. These patterns have been used with different occlusion strategies to train the LSVM classifier [24]. Later, Xiang *et al.* [15] have combined 3DVP object representation, which encodes the appearance, 3D shape, view-point, the level of occlusion and truncation, with a boosting detector based on the detection scheme in [33] in order to learn from the occluded and non-occluded 3DVPs obtained from a training set. Recently, the authors in [47] have introduced region-based features with a coordinate normalization scheme, referred to as regionlet features, and a cascaded boosting classifier to deal with the problems of detecting objects of different scales and aspect ratios. Even though these methods have been effective in dealing with these problems, they suffer from high complexity either in the training mode, as in [15, 47], or in the testing mode, as in [46].

The detection accuracy employing HOG or its variants in the spatial domain has started to saturate [4]. Recently, the fast Fourier transform (FFT) has been used with

2DHOG in order to replace the costly convolution operation in the spatial domain by multiplication in the FFT domain [48]. This scheme provides a speedup over the spatial domain counterpart. However, it is based on training an object detector in the spatial domain, which usually requires large storage and training cost.

In the *interest point-based methods*, local interest points are first detected and then described [8–11, 49, 50], followed by the construction of a codebook for objects of interest by using the features obtained from the detected image patches. For instance, Leibe *et al.* [12] use a collaboration between object detection and object probabilistic segmentation to extract features relevant to the object and the discarded background regions. In this scheme, interest points are extracted, and then an implicit shape model (ISM) is used to construct the codebook for all objects. The ISM allowed learning the object model using a few training examples. However, the codebook construction usually requires a high computational cost with large datasets.

1.2.2 Multi-Object Tracking

In the past decade, a lot of attention has been paid on detecting and tracking one or more objects in videos. Recent advancement in object detection has facilitated collaboration between the detection and tracking modules for multi-object tracking [5]. Robust multi-object tracking involves the resolution of many problems such as occlusion, appearance variation, and illumination change. A pre-trained object detector that is robust to appearance variation of one specific class is often used as a critical module of most multi-object tracking methods. Specifically, one detector encodes the generic pattern information about a certain object class (for example, cars), and a single tracker models the appearance of the specific target to maintain the target identity in an image sequence. However, an object detector is likely to generate false positives and negatives, thereby affecting the performance of the tracker in terms of data association and online model update.

In multi-object tracking, offline methods based on global optimization of all object trajectories usually perform better than their online counterparts [51–59]; an experimental evaluation of recent methods can be found in [60]. For instance, to solve the data association problem, Brendel *et al.* [52] formulated this problem as finding the maximum-weight independent set of a graph of tracklets, while Zamir *et al.* [55] used generalized minimum clique graphs. In [59], this problem was solved by using a sliding window of three frames to generate short tracklets. The minimum-cost network flow is then used to optimize the overall object trajectories. For real-time applications, online methods [5, 61–63] have been developed within the detection-based tracking framework, where the data association between detections and trackers are carried out online.

Online multi-object tracking can be carried out by using joint state-space models for multi-targets [61, 64–68]. For instance, a mixture particle filter has been proposed in [61] to compute the posterior probability via a collaboration between an object detector and the proposal distribution of the particle filter. However, the joint state-space tracking methods are of high computational complexity. The probability hypothesis density filter [69] has been incorporated in visual multi-target tracking [67, 70], since the time complexity is linear with respect to the number of targets. However, it does not maintain the target identity, and consequently, requires an online clustering method to detect the peaks of the particle weights and applies data association to each cluster.

Numerous online multi-object tracking methods have dealt with the trackers independently [5, 63, 71–73]. In [5], a method based on a particle filter and two human detectors with different features was developed, where the observation model depends on the associated detection, the detector confidence density and the likelihood of appearance. In addition, Shu *et al.* [63] introduced a part-based pedestrian detector for online multi-person tracking. These methods are likely to have low recall as the detector and tracker are not integrated within the same framework.

Particle filters suffer from the degeneracy problem [74], wherein after a few iterations, except for one particle all the others have negligible weights. This problem has been addressed by several authors [75–78] with effective proposal distributions and re-sampling steps. Rui and Chen [76] used the unscented Kalman filter for generating the proposal distribution, while Han *et al.* [79] used a genetic algorithm to increase the diversity of the particles. Recently, the Metropolis Hastings algorithm [80] has been used to sample particles from associated detections in the detection-based tracking framework [78]. The above-mentioned methods do not exploit collaboration between detectors and trackers [76, 79], nor do they consider the effect of false positive detections on the trackers [78].

1.3 Motivation

Detection accuracy of object detectors that are based on features obtained in the spatial domain has started to saturate [4]. Not much effort has been made in using transform-domain features with a view of improving the accuracy of object detection or reducing the storage and training cost. An object detector usually suffers from false positives and missing detections which affect the tracking process, when used for tracking. In view of this, a careful study is needed to develop a more effective collaborative model between detections and trackers in order to improve the tracking process.

1.4 Objectives and Organization of the Thesis

The objective of this thesis is two fold: (i) to establish a relationship between the TD2DHOG features obtained at two different resolutions and use this relationship in developing a vehicle detection scheme that is able to tackle the problems of variations in the vehicle scale and view, and changes in illumination and environmental

conditions at low training and storage costs, and (ii) to develop a robust and efficient online detection-based multi-object tracking scheme.

In the first part of the thesis, a new vehicle detection scheme using transform-domain 2DHOG features is proposed [81, 82]. The method is based on extracting the 2DHOG features from the input image and then applying 2D discrete Fourier or cosine transform to these 2DHOG features. This is followed by a truncation process through which only the low frequency coefficients, referred to as the transform-domain 2DHOG (TD2DHOG) features, are retained. It is shown that the TD2DHOG features obtained from an image at the original resolution and a downsampled version from the same image are approximately the same within a multiplicative factor. This property is then utilized in developing a scheme for the detection of vehicles of various resolutions using a single classifier rather than multiple resolution-specific classifiers. Extensive experiments are conducted on three vehicle detection datasets, namely, *UIUC car detection dataset* [83], the *USC multi-view car detection dataset* [28], and the *LISA 2010 dataset* [84]; the results show that the use of the single classifier in the proposed detection scheme reduces drastically the training and storage costs over the use of a classifier pyramid, yet providing a detection accuracy similar to that obtained using TD2DHOG features with a classifier pyramid. Furthermore, the proposed method provides a detection accuracy that is similar to or even better than that provided by the state-of-the-art techniques.

In the second part of the thesis, a robust collaborative model that enhances the interaction between a pre-trained object detector and a number of particle filter-based single-object online trackers is presented [85, 86]. For each frame, an association between a detection and a tracker is constructed. For each tracker, a motion model that incorporates the associated detections with the object dynamics, and a likelihood function that provides different weights for the propagated particles and the newly created ones sampled from the associated detections are introduced, with a view to reduce the effect of detection errors on the tracking process. Finally, a new image

sample selection scheme is introduced in order to update the appearance model of a given tracker. Extensive experiments are conducted on seven challenging sequences, namely, the *PETS09-S2L1, PETS09-S2L2* [87], *UCF Parking Lot (UCF-PL)* dataset [63], *Soccer* dataset [62], *Town Center* dataset [88], and *Urban* as well as *Sunny* sequences from *LISA 2010* dataset [84], which show that the proposed scheme generally outperforms state-of-the-art methods.

The organization of the thesis is as follows: In Chapter 2, we present a brief overview on 2DHOG features, and the effect of image resampling on the 2DHOG features. In Chapter 3, TD2DHOG features are defined and a method of extracting the TD2DHOG features is presented. It is shown that the TD2DHOG features obtained from an image at the original resolution and a downsampled version from the same image are approximately the same within a multiplicative factor. A model for the multiplicative factor has been proposed and the parameters for this model are determined using various vehicle detection datasets. Then, this model is used in proposing a scheme for vehicle detection of different resolutions using a single classifier rather than a classifier pyramid. Extensive experiments are conducted to study the performance of the proposed scheme for vehicle detection. In Chapter 4, a robust online multi-object tracking scheme in the particle filter framework is presented. In this scheme, a robust collaborative model for the interaction between a number of single-object online trackers and a pre-trained object detector is presented. A novel image sample selection scheme is introduced to update each tracker by using relevant samples from its trajectory. Also, a data association method with partial occlusion handling by using diverse generative models composed of sparsity-based generative model, and two-dimensional principal component analysis (2DPCA) generative model is presented. Extensive experiments are conducted to study the effectiveness of the proposed scheme in enhancing the multi-object tracking performance. Chapter 5 concludes with the highlights of the contributions of the thesis, followed by some suggestions for future work.

Chapter 2

Review of Background Material

In this chapter, we present a brief review of the material required for the development of the proposed object detection and tracking schemes in subsequent chapters.

2.1 Two-Dimensional HOG Features

Two-dimensional histogram of oriented gradients (2DHOG) features are similar to the HOG features introduced by Dalal and Triggs [6], the difference being the way in which the features are represented, namely, in a 2D matrix format in the case of the former and a 1D vector format in the case of the latter. The 2DHOG features have been used in a number of papers [22–24].

Figure 2.1 shows block diagram for the extraction process of 2DHOG features from an input car image of size (32×96). Let us consider an image, $\mathbf{I}$, of size $(M_1 \times M_2)$, and divide it into non-overlapping cells of size $(\eta_1 \times \eta_2)$ pixels. The 2DHOG features are computed from the input image as follows. First, we convolve the image $\mathbf{I}$ with the filter $L = [-1, 0, 1]$ and its transpose $L^\top$ to obtain the gradients $g_x(i, j)$ and $g_y(i, j)$, in the x and y directions, respectively, where i and j denote the pixel indices. Then, we compute the magnitude $\Gamma(i, j)$ and the orientation $\theta(i, j)$ of the gradient at (i, j)

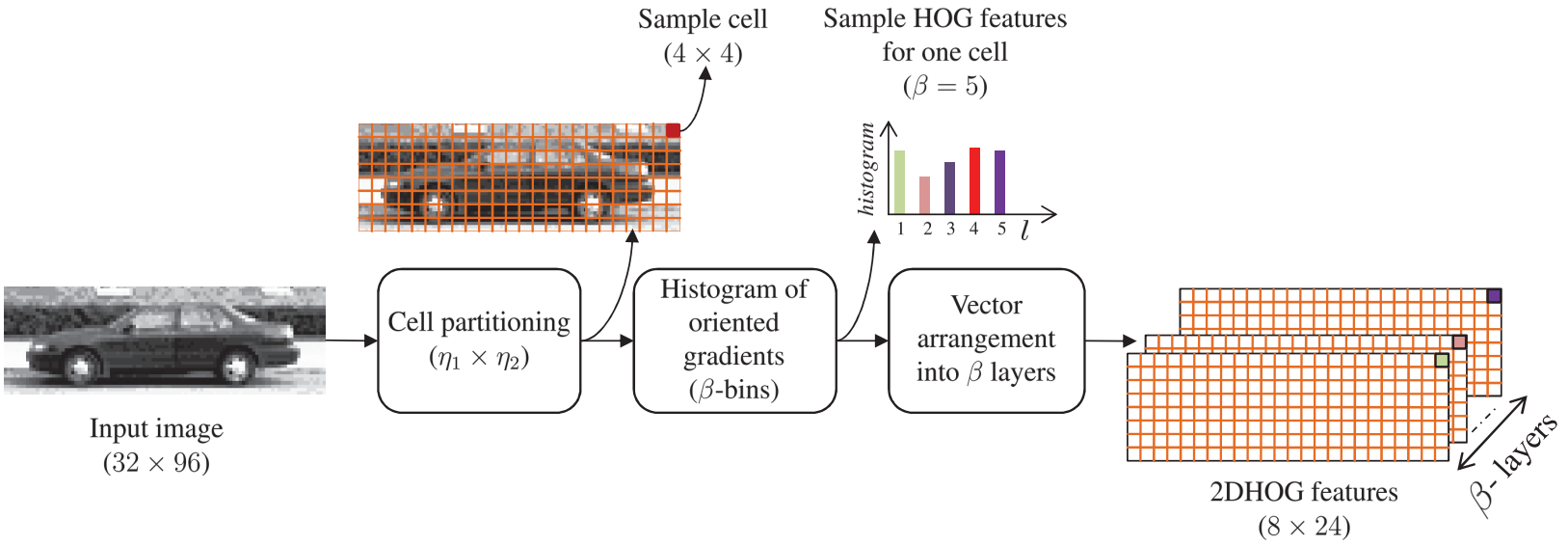


Figure 2.1: Block diagram for the process of extracting 2DHOG features from an input image of size (32×96), where $M_1 = 32, M_2 = 96$, $\eta_1 = \eta_2 = 4$ and $\beta = 5$.

as

$$\begin{aligned}\Gamma(i,j) =& \sqrt{g_x(i,j)^2 + g_y(i,j)^2} \\ \theta(i,j) =& \arctan\left(g_y(i,j)/g_x(i,j)\right)\end{aligned} \tag{2.1}$$

Next, the orientation $\theta(i,j)$ is quantized into β bins such that the quantized orientation $\hat{\theta}(i,j) \in \Omega$ and $\Omega = [0, \pi/\beta, ..., (\pi - \pi/\beta)]$. Then, the 2DHOG features for the l^{th} layer, $h^l(\hat{i},\hat{j})$, can be computed using the equation

$$\mathbf{h}^l(\hat{i},\hat{j}) = \sum_{i=(\hat{i}-1)\eta_1+1}^{\hat{i}\eta_1} \left(\sum_{j=(\hat{j}-1)\eta_2+1}^{\hat{j}\eta_2} \Gamma(i,j)\delta_l(i,j) \right) \tag{2.2}$$

where

$$\delta_l(i,j) = \begin{cases} 1, & \text{if } \hat{\theta}(i,j) = \Omega(l) \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

$\hat{i}$ and $\hat{j}$ being the cell indices, $1 \leq \hat{i} \leq \tilde{M}_1 = M_1/\eta_1$, $1 \leq \hat{j} \leq \tilde{M}_2 = M_2/\eta_2$, such that $\tilde{M}_1$ and $\tilde{M}_2$ are integers. Thus, the 2D representation for the HOG features results in β-layers, $\mathbf{h}^l\,(l = 1, 2, ..., \beta)$, where the spatial relation between neighboring cells is maintained, and the size of each layer is $(\tilde{M}_1 \times \tilde{M}_2)$.

2.2 Effect of Image Resampling on Channel Features

Statistics of resampled images in the spatial domain have been studied in [89, 90]. Recently, the effect of image resampling on 2D channel features in the spatial domain, such as color image, gradient magnitude and 2DHOG, has been studied by Dollár *et al.* in [32, 33]. In this section, we give a brief description of the work in [33], which will be used later in developing the proposed detection scheme in the subsequent chapter.

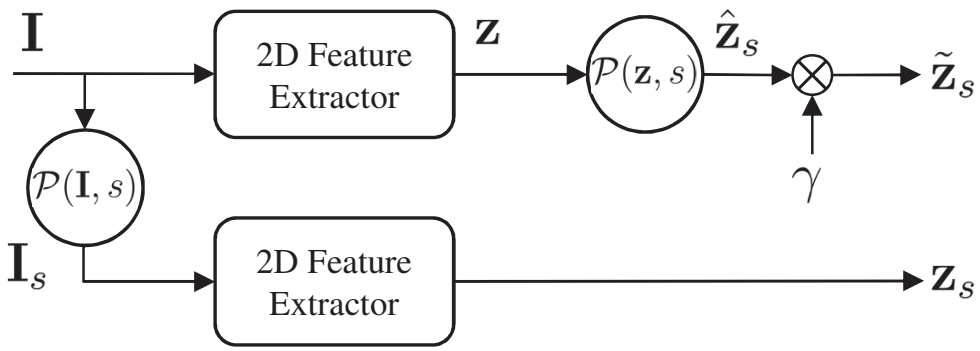


Figure 2.2: Block diagram illustrating the approximate relationship between the resampled features of an image at a given resolution and the features extracted from a resampled version of the same image.

Let $\mathbf{I}_s = \mathcal{P}(\mathbf{I}, s)$ denote the input image $\mathbf{I}$ resampled by a factor s, where $s < 1$ represents downsampling, $s > 1$ represents upsampling, and $\mathcal{P}$ represents the resampling operator in the spatial domain. The exact channel features extracted from the image at the original resolution, and the same image at a different resolution can be represented by $\mathbf{z} = \Lambda(\mathbf{I})$, and $\mathbf{z}_s = \Lambda(\mathbf{I}_s)$, respectively, where Λ denotes a 2D spatial-domain feature extractor. It has been shown in [33] that resampling the image $\mathbf{I}$ by a factor s, $\mathbf{I}_s = \mathcal{P}(\mathbf{I}, s)$, followed by computing the exact 2D channel features, $\mathbf{z}_s = \Lambda(\mathbf{I}_s)$, can be approximated by resampling the feature channel, $\mathbf{z}$, followed by a multiplicative factor, γ, that is modeled by using the power law [33] as

$$\mathbf{z}_s = \Lambda(\mathcal{P}(\mathbf{I}, s)) \approx \tilde{\mathbf{z}}_s = \gamma \mathcal{P}(\mathbf{z}, s) \tag{2.4}$$

where

$$\gamma = a_0 s^{-\lambda} \tag{2.5}$$

and a_0 and λ depend on the channel type, and are empirically determined. This relationship is illustrated by the block diagram of Figure 2.2. The values of a_0 and λ are not necessarily the same for the case of upsampling and downsampling for the

same channel type.

For object detection using a single detection window, one constructs an image pyramid encompassing different scales, and then extracts the features from every scale in the pyramid. The use of the approximation in (2.4) allows the features generated at one scale from the image pyramid to approximate the features at nearby scales, thus reducing the cost of feature computation.

2.3 Summary

In this chapter, background material that is required for the investigation carried out in the succeeding chapters has been described. First, a brief description of 2DHOG features has been presented. Then, the work done by Dollár *et al.* in [33] to study the effect of image resampling on the channel features in the spatial domain has been briefly discussed.

Chapter 3

Transform-Domain Two-Dimensional HOG Feature and its use in Vehicle Detection

In this chapter, we introduce the concept of transform-domain 2DHOG features and use it to propose a new vehicle detection scheme [81, 82]. In Section 3.1, we study the effect of downsampling a grayscale image on its DFT and DCT versions. In Section 3.2, transform-domain 2DHOG (TD2DHOG) features are defined and a method of extracting these features is presented. A relationship between the TD2DHOG features obtained from an image at the original resolution and a downsampled version from the same image is established. In Section 3.3, we use this relationship in proposing a scheme for vehicle detection of different resolutions using a single classifier rather than a classifier pyramid. In Section 3.4, the performance of the proposed vehicle detection scheme is studied by carrying out extensive experiments using a number of publicly available vehicle detection datasets and compared with that of the state-of-the-art techniques. Finally, Section 3.5 summarizes the work of this chapter.

3.1 Effect of Downsampling a Grayscale Image on its Transformed Version

In this section, we study the effect of downsampling a grayscale image on its DFT and DCT versions, and these results are then used in Section 3.2 to investigate the effect of image downsampling on transform-domain 2DHOG features.

3.1.1 Effect on the DFT Version

Let the N-point 1DDFT for the discrete time sequence, $z[n] \in \mathbb{R}$, be denoted as $Z_N[k]$, where $n = 0, 1, ..., N-1$, $k = 0, 1, ..., N-1$, N is an even integer multiple of K, and K being an integer. Let an ideal low pass filter of unity gain and a cutoff frequency $N_c \leq N/(2K)$ be used in order to bandlimit the signal. By downsampling z by K in the time domain, the downsampled signal $\hat{z}$ of length $\hat{N} = N/K$ is obtained. Then, the $\hat{N}$-point 1DDFT is employed on the downsampled signal, $\hat{z}$, in order to obtain the downsampled signal in the frequency domain, $\hat{Z}_{\hat{N}}$. Now, the relations between the original signal and its downsampled version in the time domain and that in the frequency domain are given by

$$\hat{z}[n] = z[Kn] \tag{3.1}$$

$$\hat{Z}_{\hat{N}}[k] = \frac{1}{K} \sum_{i=0}^{K-1} Z_N \left[k + i\hat{N} \right] \tag{3.2}$$

where $n = 0, 1, ..., \hat{N}-1$, and $k = 0, 1, ..., \hat{N}-1$. It is clear from (3.2) that the downsampled signal in the 1DDFT domain, $\hat{Z}_{\hat{N}}$, is represented by a sum of K shifted copies of the original signal in the 1DDFT domain, Z_N, scaled by the factor $1/K$ [91]. Figure 3.1 illustrates an example of this in the DFT domain, when $N = 16$, $\hat{N} = 8$, $K = 2$, and $N_c = 4$. Since the original signal is bandlimited, then for

$k = 0, 1, ..., c_1 - 1$, $c_1 \leq N_c$, the contribution of the summation shown in (3.2) is only coming from the first copy of Z_N at $i = 0$, and so we have

$$Z_N[k] = K\hat{Z}_{\hat{N}}[k] \tag{3.3}$$

This result confirms that given in [92].

We now consider a 2D signal. Let $g \in \mathbb{R}^2$ represent a grayscale image in the spatial domain of size $(N_1 \times N_2)$, where N_1 and N_2 are even integer multiples of K_1 and K_2, respectively, K_1 and K_2 being integers. Assume that an ideal low pass filter of unity gain and cutoff frequencies $N_{c_1} \leq N_1/(2K_1)$ and $N_{c_2} \leq N_2/(2K_2)$ is used to bandlimit the original signal. Downsampling g by a factor K_1 in the y direction, and K_2 in the x direction results in $\hat{g}[n, m] = g[K_1 n, K_2 m]$ of size $(\hat{N}_1 \times \hat{N}_2)$, where n and m represent the spatial domain discrete sample indices, $0 \leq n \leq \hat{N}_1 - 1$, $0 \leq m \leq \hat{N}_2 - 1$, $\hat{N}_1 = N_1/K_1$ and $\hat{N}_2 = N_2/K_2$. We now take the 2DDFT of g and $\hat{g}$ to obtain G_{N_1,N_2} and $\hat{G}_{\hat{N}_1,\hat{N}_2}$ corresponding to the 2DDFT coefficients of the original image and that of its downsampled version, respectively. Similar to the case of 1DDFT, the relation between $G_{N_1,N_2}[u, v]$ and $\hat{G}_{\hat{N}_1,\hat{N}_2}[u, v]$ can be expressed as

$$\hat{G}_{\hat{N}_1,\hat{N}_2}[u, v] = \frac{1}{K_1 K_2} \sum_i \sum_j G_{N_1,N_2}[u + i\hat{N}_1, v + j\hat{N}_2] \tag{3.4}$$

where $u = 0, 1, ..., \hat{N}_1 - 1$, $v = 0, 1, ..., \hat{N}_2 - 1$, $i = 0, 1, ..., K_1 - 1$, and $j = 0, 1, ..., K_2 - 1$. It is seen from this equation that the downsampled image in the 2DDFT domain is represented by a sum of $K_1 \times K_2$ shifted copies of the original image in the 2DDFT domain and scaled by the factor $1/(K_1 K_2)$. Let c_1 and c_2 denote the maximum frequencies retained by the truncation operator. For $u = 0, 1, ..., c_1 - 1$, $v = 0, 1, ..., c_2 - 1$, $c_1 \leq N_{c_1}$, and $c_2 \leq N_{c_2}$ the contribution of the summation shown in (3.4) is from the copy corresponding to $i = j = 0$, and we can obtain the following relation

$$G_{N_1,N_2}[u, v] = K_1 K_2 \hat{G}_{\hat{N}_1,\hat{N}_2}[u, v] \tag{3.5}$$

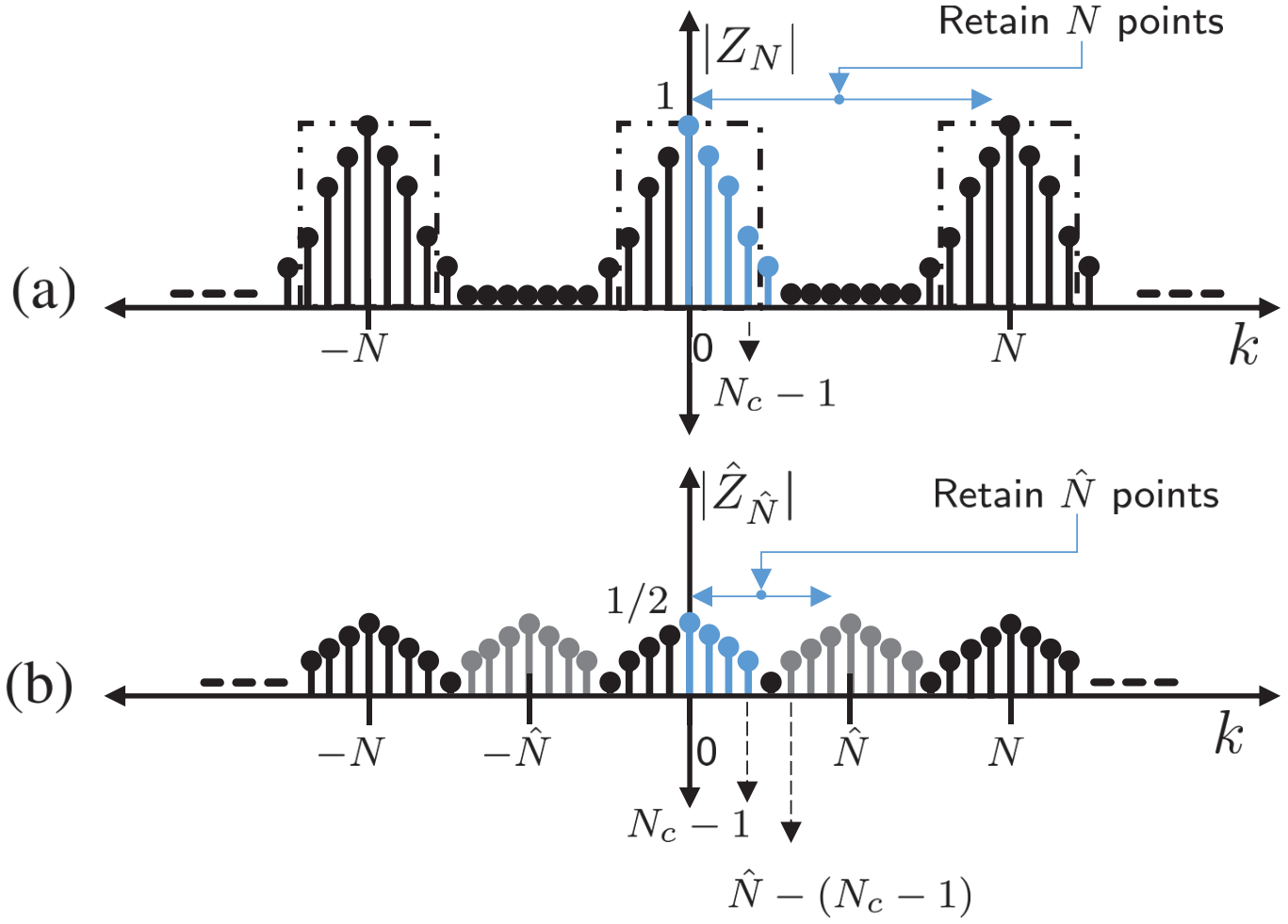


Figure 3.1: (a) Magnitude of a signal in the DFT domain $Z_N[k]$, where a low pass filter with cutoff frequency N_c is used to bandlimit the signal. (b) Magnitude of the downsampled signal in the DFT domain $\hat{Z}_{\hat{N}}[k]$, where $N = 16, K = 2, \hat{N} = 8$, and $N_c = 4$.

From the above equation it is seen that the ratio between a grayscale image in the 2DDFT domain and that of its downsampled version is $K_1 K_2$.

3.1.2 Effect on the DCT Version

In [93] the N-point 1DDCT, X_N, for the discrete time sequence, $x \in \mathbb{R}$, is given by

$$X_N[k] = \hat{\Gamma}_N[k] \sum_{n=0}^{N-1} x[n] \cos \frac{\pi(2n+1)k}{2N} \tag{3.6}$$

where $\hat{\Gamma}_N[k] = \sqrt{1/N}$ for $k = 0$, and $\hat{\Gamma}_N[k] = \sqrt{2/N}$ for $0 < k \leq N-1$. The N-point 1DDCT can be computed by 2N-point 1DDFT for a sequence, $y[n]$, as follows. First, let $x[n]$ be a bandlimited signal and $y[n]$ be defined as

$$y[n] = \begin{cases} x[n], & 0 \leq n \leq N-1 \\ 0, & N \leq n \leq 2N-1 \end{cases} \tag{3.7}$$

The 1DDFT is employed on y in order to obtain Y_{2N}. It has been shown in [93] that the signal $X_N[k]$ in the 1DDCT domain is related to $Y_{2N}[k]$ by

$$X_N[k] = \hat{\Gamma}_N[k] \mathrm{Re}(Y_{2N}[k] e^{-j\frac{\pi k}{2N}}) \tag{3.8}$$

where $k = 0, 1, ..., N-1$, and Re() is a function which returns the real part of an input complex number. Let an ideal low pass filter of gain unity and a cutoff frequency $N_c \leq N/K$ be used in order to bandlimit the signal Y_{2N}, where N is an even integer multiple of K, and K being an integer. Let $E_{2N}[k]$ be a 1D signal in the 1DDFT domain, and be defined as $E_{2N}[k] = Y_{2N}[k] e^{-j\frac{\pi k}{2N}}$. From the downsampling theorem given by (3.2), downsampling $E_{2N}[k]$ by a factor K in the 1DDFT domain is obtained as:

$$\hat{E}_{2\hat{N}}[k] = \frac{1}{K} \sum_{i=0}^{K-1} E_{2N} \left[k + i2\hat{N} \right] \tag{3.9}$$

where $\hat{E}_{2\hat{N}}$ is of length $2\hat{N} = 2N/K$, and $k = 0, 1, ..., N-1$. Figures 3.2 (a) and (b) illustrate an example for $E_{2N}[k]$ and $\hat{E}_{2\hat{N}}[k]$, respectively, where $N = 8, K = 2, \hat{N} = 4$, and $N_c = 4$. Now, the downsampled signal in the 1DDCT domain, $\hat{X}_{\hat{N}}$ of length $\hat{N}$, can be obtained as follows:

$$\hat{X}_{\hat{N}}[k] = \hat{\Gamma}_{\hat{N}}[k]\mathrm{Re}(\hat{E}_{2\hat{N}}[k]) \tag{3.10}$$

$$= \hat{\Gamma}_{\hat{N}}[k]\mathrm{Re}(\frac{1}{K}\sum_{i=0}^{K-1} Y_{2N}[k + i2\hat{N}]e^{-j\frac{\pi(k+i2\hat{N})}{2N}}) \tag{3.11}$$

Let c_1 denote the maximum frequency retained by the truncation operator. Since Y_{2N} is bandlimited to the maximum frequency $N_c \leq N/K$, then for $k = 0, 1, ..., c_1 - 1$, where $c_1 \leq N_c$, the contribution of the summation shown in (3.11) is coming only from $i = 0$ copy, and so we can simplify the above relation as

$$\hat{X}_{\hat{N}}[k] = \frac{1}{K}\hat{\Gamma}_{\hat{N}}[k]\mathrm{Re}(Y_{2N}[k]e^{-j\frac{\pi k}{2N}}) \tag{3.12}$$

$$= \frac{\hat{\Gamma}_{\hat{N}}[k]}{K\hat{\Gamma}_N[k]}\hat{\Gamma}_N[k]\mathrm{Re}(Y_{2N}[k]e^{-j\frac{\pi k}{2N}}) = \frac{\sqrt{1/\hat{N}}}{K\sqrt{1/N}}X_N[k] \tag{3.13}$$

$$= \frac{1}{\sqrt{K}}X_N[k] \tag{3.14}$$

Thus, the relation between a 1DDCT transformed signal and its downsampled version in the 1DDCT domain can be expressed as

$$X_N[k] = \sqrt{K}\hat{X}_{\hat{N}}[k] \tag{3.15}$$

where $0 \leq k \leq c_1 - 1$.

We now extend the above result for 1DDCT to the case of 2DDCT. The 2DDCT

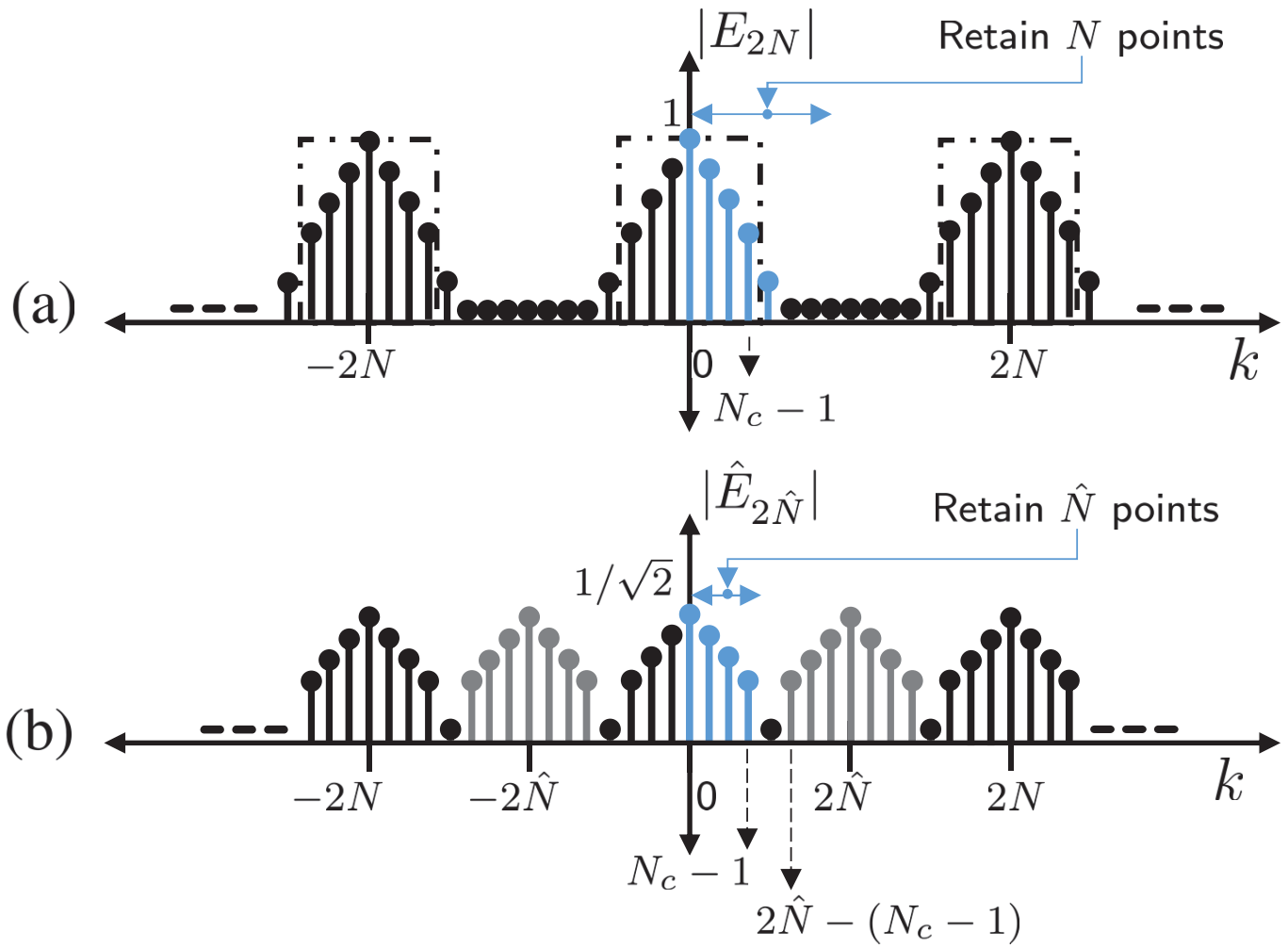


Figure 3.2: (a) Magnitude of the signal $E_{2N}[k]$ defined as $E_{2N}[k] = Y_{2N}[k]e^{-j\frac{\pi k}{2N}}$, where a low pass filter with cutoff frequency N_c is used to bandlimit the signal. (b) Magnitude of the downsampled signal in the DFT domain $\hat{E}_{2\hat{N}}[k]$, where $N = 8, K = 2, \hat{N} = 4$, and $N_c = 4$.

for a grayscale image in the spatial domain, $g \in \mathbb{R}^2$, is given by

$$\begin{aligned} G_{N_1,N_2}[u,v] =& \hat{\Gamma}_{N_1}[u]\hat{\Gamma}_{N_2}[v] \sum_{m=0}^{N_2-1} \sum_{n=0}^{N_1-1} g[n,m] \cos\left(\frac{\pi(2n+1)u}{2N_1}\right) \\ & \times \cos\left(\frac{\pi(2m+1)v}{2N_2}\right) \end{aligned} \tag{3.16}$$

where $0 \leq u \leq N_1-1$, $0 \leq v \leq N_2-1$, $\hat{\Gamma}_{N_1}[k] = \sqrt{1/N_1}$ for $k = 0$ and $\hat{\Gamma}_{N_1}[k] = \sqrt{2/N_1}$ for $0 < k \leq N_1-1$. Let N_1 and N_2 be even multiples of K_1, and K_2, respectively, where K_1 and K_2 are the downsampling factors in the y and the x directions, respectively. Let $g[n,m]$ be a bandlimited signal, and the signal $a \in \mathbb{R}^2$, of size $(2N_1 \times 2N_2)$, be defined as

$$a[n,m] = \begin{cases} g[n,m], & 0 \leq n \leq N_1-1, 0 \leq m \leq N_2-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

The $N_1 \times N_2$-point 2DDCT can be computed by $2N_1 \times 2N_2$-point 2DDFT for a signal, $a[n,m]$, as follows. First, the 2DDFT is employed on $a[n,m]$ in order to obtain $A_{2N_1,2N_2}$. Similar to the 1DDCT case, the relation between the signal in the 2DDCT domain $G_{N_1,N_2}[u,v]$, and $A_{2N_1,2N_2}[u,v]$ can be expressed as

$$G_{N_1,N_2}[u,v] = \hat{\Gamma}_{N_1}[u]\hat{\Gamma}_{N_2}[v]\mathrm{Re}(A_{2N_1,2N_2}[u,v]e^{-j(\frac{\pi u}{2N_1}+\frac{\pi v}{2N_2})}) \tag{3.18}$$

where $0 \leq u \leq N_1-1$, $0 \leq v \leq N_2-1$. Let c_1, c_2 denote the maximum frequencies retained by the truncation operator, where $c_1 < \hat{N}_1$, $c_2 < \hat{N}_2$, $\hat{N}_1 = N_1/K_1$, and $\hat{N}_2 = N_2/K_2$. Assume $A_{2N_1,2N_2}$ is bandlimited to the maximum frequencies $(\hat{N}_1, \hat{N}_2)$. Then, the downsampled signal in the 2DDCT domain, $\hat{G}_{\hat{N}_1,\hat{N}_2}$, can be obtained as

$$\begin{aligned} \hat{G}_{\hat{N}_1,\hat{N}_2}[u,v] &= \frac{1}{K_1K_2}\hat{\Gamma}_{\hat{N}_1}[u]\hat{\Gamma}_{\hat{N}_2}[v]\mathrm{Re}(A_{2N_1,2N_2}[u,v]e^{-j(\frac{\pi u}{2N_1}+\frac{\pi v}{2N_2})}) \\ &= \frac{\hat{\Gamma}_{\hat{N}_1}[u]\hat{\Gamma}_{\hat{N}_2}[v]}{K_1K_2\hat{\Gamma}_{N_1}[u]\hat{\Gamma}_{N_2}[v]}\hat{\Gamma}_{N_1}[u]\hat{\Gamma}_{N_2}[v]\mathrm{Re}(A_{2N_1,2N_2}[u,v] \\ &\quad \times e^{-j(\frac{\pi u}{2N_1}+\frac{\pi v}{2N_2})}) \\ &= \frac{1}{\sqrt{K_1K_2}}G_{N_1,N_2}[u,v] \end{aligned} \tag{3.19}$$

where $0 \leq u \leq c_1 - 1$ and $0 \leq v \leq c_2 - 1$. Thus, the relation between the 2DDCT coefficients of the original image and that of the downsampled version is given by

$$G_{N_1,N_2}[u,v] = \sqrt{K_1 K_2}\hat{G}_{\hat{N}_1,\hat{N}_2}[u,v] \tag{3.20}$$

where $\hat{N}_1 = N_1/K_1$, $\hat{N}_2 = N_2/K_2$, $u = 0, 1, ..., c_1 - 1$, $v = 0, 1, ..., c_2 - 1$, $c_1 \leq N_1/K_1$, and $c_2 \leq N_2/K_2$.

3.2 Transform-Domain 2DHOG Features

In this section, we first define 2DHOG features in the transform domain. Then, utilizing the results derived in Section 3.1, we investigate the relationship between the transform-domain 2DHOG features obtained from an image of a given resolution and those obtained from a downsampled version of the same image.

3.2.1 Extraction of TD2DHOG Features

Consider an input image $\mathbf{I}$ of size $(M_1 \times M_2)$. Let it be divided into non-overlapping cells of size $(\eta_1 \times \eta_2)$, where M_1 and M_2 are integer multiples of powers of 2, and η_1 and η_2 are integer powers of 2. Now, 2DHOG features are computed by following the steps explained in Section 2.1, resulting in β layers, where each layer corresponds to a certain quantized gradient orientation from 0° to 180°. The 2DHOG features of the l^{th} layer, denoted by $\mathbf{h}^l$, is of size $(\tilde{M}_1 \times \tilde{M}_2)$, $\tilde{M}_1$ and $\tilde{M}_2$ being integer multiples of powers of 2. Each 2DHOG layer, $\mathbf{h}^l$, is partitioned into a number of non-overlapping blocks, N_x and N_y in the x and y directions, respectively, where N_x and N_y are integers. Let $\mathbf{h}^l_{\imath\jmath}$, of size $(b \times b)$, represent the 2DHOG features of the $(\imath, \jmath)^{th}$ block of the l^{th} layer, where $1 \leq \imath \leq N_y$, $1 \leq \jmath \leq N_x$, b being an integer power of 2. The

block-partitioned 2DHOG features in the l^{th} layer can be represented as

$$\mathbf{h}^l = \begin{bmatrix} \mathbf{h}^l_{11} & ... & \mathbf{h}^l_{1N_x} \\ \vdots & \ddots & \vdots \\ \mathbf{h}^l_{N_y1} & ... & \mathbf{h}^l_{N_yN_x} \end{bmatrix} \tag{3.21}$$

This block partitioning is known to offer a robustness to partial occlusion [14, 21]. To illustrate let us consider an image of size 32×96, a cell size of 4×4, and $\beta = 5$. If $b = 8$, then $N_x = \tilde{M}_2/b = M_2/(\eta_2 b) = 3$, and $N_y = \tilde{M}_1/b = M_1/(\eta_1 b) = 1$. Hence, each of the five layers is partitioned into 3 blocks of size 8×8. However, if $b = 4$, then $N_x = 6$ and $N_y = 2$; that is, each of the layers is partitioned into 12 blocks of size 4×4.

Next, we apply the appropriate 2D transform, 2DDFT or 2DDCT, on each block resulting in 2DHOG of the corresponding block in the transform domain. Let $\mathbf{H}^l_{ij} = T(\mathbf{h}^l_{ij})$, where $T(.)$ represents the transform. The corresponding 2DHOG features in the transform domain can be represented as

$$\mathbf{H}^l = \begin{bmatrix} \mathbf{H}^l_{11} & ... & \mathbf{H}^l_{1N_x} \\ \vdots & \ddots & \vdots \\ \mathbf{H}^l_{N_y1} & ... & \mathbf{H}^l_{N_yN_x} \end{bmatrix} \tag{3.22}$$

Let $\boldsymbol{\phi}_{c_1c_2}(.)$ denote the 2D truncation operator in the transform domain that truncates the coefficients corresponding to the frequencies greater than the frequencies c_1 and c_2. By applying $\boldsymbol{\phi}_{c_1c_2}(.)$ on each block, $\mathbf{H}^l_{ij}$, we can obtain the truncated features as $\hat{\mathbf{H}}^l_{ij} = \boldsymbol{\phi}_{c_1c_2}(\mathbf{H}^l_{ij})$ of size $(c_1 \times c_2)$. Then, these features can be represented as

$$\hat{\mathbf{H}}^l = \begin{bmatrix} \hat{\mathbf{H}}^l_{11} & ... & \hat{\mathbf{H}}^l_{1N_x} \\ \vdots & \ddots & \vdots \\ \hat{\mathbf{H}}^l_{N_y1} & ... & \hat{\mathbf{H}}^l_{N_yN_x} \end{bmatrix} \tag{3.23}$$

where the size of $\hat{\mathbf{H}}^l$ is $(\hat{M}_1 \times \hat{M}_2)$, $\hat{M}_1 = c_1N_y$ and $\hat{M}_2 = c_2N_x$. We call the above

truncated transform-domain 2DHOG features given by $\hat{\mathbf{H}}^l$ as TD2DHOG features. We refer to the TD2DHOG features as DFT2DHOG and DCT2DHOG features when the 2D transform used is 2DDFT and 2DDCT, respectively. The scheme for obtaining the DCT2DHOG features is illustrated in Figure 3.3 for an image of size 32×96 with a cell size of 4×4, $\beta = 5$, and 2DDCT is employed with block size $b = 8$, and $c_1 = c_2 = 4$. It is noted that for this example the size of $\hat{\mathbf{H}}^l$ is 4×12.

3.2.2 Effect of Image Downsampling on TD2DHOG Features

In Section 3.1, we obtained the relation between the original image and its downsampled version when they are transformed by 2DDFT or 2DDCT. Now, in order to study the effect of image downsampling on the features in the transform domain, we use the block diagram shown in Figure 3.4. For the original image $\mathbf{I}$, a 2DHOG feature extraction operator $\Lambda(.)$ is employed to obtain $\mathbf{z} = \Lambda(\mathbf{I})$. Then, we apply to $\mathbf{z}$ an appropriate 2D transform (2DDFT or 2DDCT), with a block size $b \times b$, followed by a truncation operation retaining the $c \times c$ low frequency coefficients for each block. The TD2DHOG features so obtained are denoted by $\hat{\mathbf{Z}} = \hat{T}(\mathbf{z})$, where $\hat{T}$ represents the transform operation followed by the truncation operation. Let $\mathbf{I}_{1/K}$ denote the image $\mathbf{I}$ downsampled by a factor K in both the x and y directions. Since $\mathbf{I}_{1/K} = \mathcal{P}(\mathbf{I}, 1/K)$, $\mathcal{P}$ representing the downsampling operator, the features extracted from the downsampled image is given by $\mathbf{z}_{1/K} = \Lambda(\mathcal{P}(\mathbf{I}, 1/K))$. We now obtain the features $\hat{\mathbf{Z}}_{1/K} = \hat{T}_{1/K}(\mathbf{z}_{1/K})$ in the transform domain, where the features $\mathbf{z}_{1/K} = \Lambda(\mathbf{I}_{1/K})$, and $\hat{T}_{1/K}$ represents the transform operation with a block size $(b/K) \times (b/K)$ followed by the truncation operation to retain the $(c \times c)$ low frequency coefficients.

The relationship between the transform coefficients of the features obtained from the image at the original resolution $\hat{\mathbf{Z}}$ and that of its downsampled version $\hat{\mathbf{Z}}_{1/K}$ can now be obtained as follows. Equations (2.4) and (2.5) are now used to approximate $\mathbf{z}_{1/K}$ as

$$\mathbf{z}_{1/K} \approx \mathcal{P}(\mathbf{z}, 1/K) a_0' K^{\lambda} \tag{3.24}$$

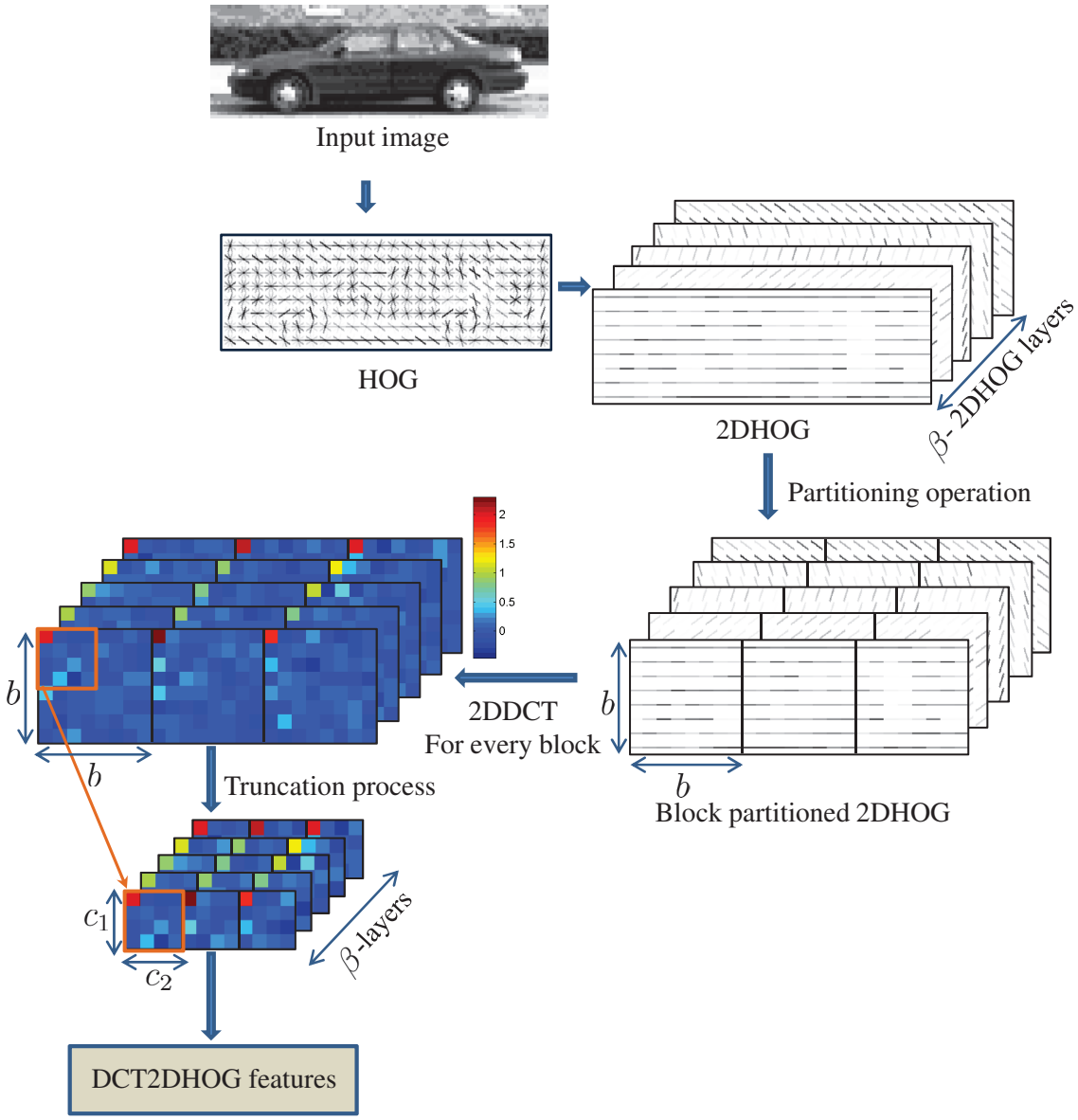


Figure 3.3: Scheme for obtaining the DCT2DHOG features for an input car image of size 32×96 using $\beta = 5$, cell size 4×4, 2DDCT block size $b = 8$ and $c_1 = c_2 = 4$.

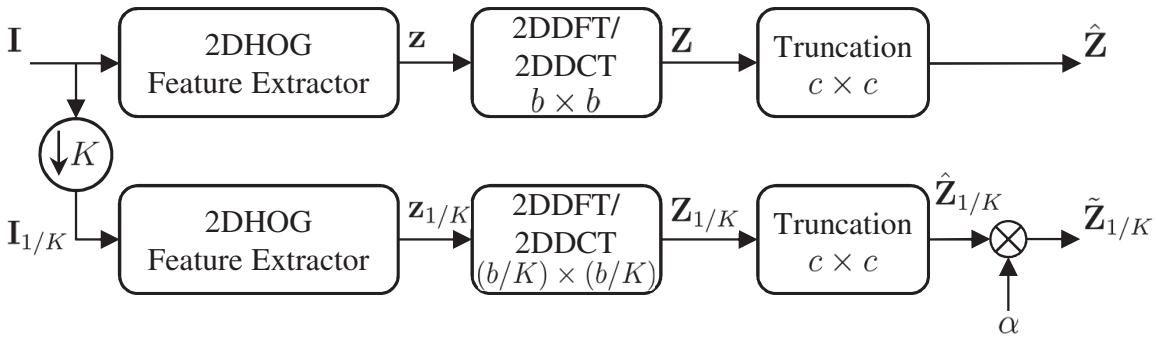


Figure 3.4: Block diagram showing the effect of downsampling an input image by an integer factor K in both the x and y directions on the transform-domain 2DHOG features, where α is a multiplicative factor that allows the features extracted from the lower resolution image to approximate the features extracted from the image at the original resolution.

where a_0' and λ are computed empirically for each channel type. Next, performing the transform operation $\hat{T}_{1/K}$ on both sides of (3.24), we obtain

$$\hat{T}_{1/K}(\mathbf{z}_{1/K}) \approx \hat{T}_{1/K}(\mathcal{P}(\mathbf{z}, 1/K))a_0' K^{\lambda}$$

i.e.,

$$\hat{\mathbf{Z}}_{1/K} \approx \hat{T}_{1/K}(\mathcal{P}(\mathbf{z}, 1/K))a_0' K^{\lambda} \tag{3.25}$$

Then, the ratio between the features in the transform domain obtained from the original image and its resampled version is

$$\frac{\hat{\mathbf{Z}}}{\hat{\mathbf{Z}}_{1/K}} \approx \frac{1}{a_0' K^{\lambda}} \times \frac{\hat{T}(\mathbf{z})}{\hat{T}_{1/K}(\mathcal{P}(\mathbf{z}, 1/K))} \tag{3.26}$$

where the first term, $1/(a_0' K^{\lambda})$, represents the power law effect, while the second term, $\hat{T}(\mathbf{z})/\hat{T}_{1/K}(\mathcal{P}(\mathbf{z}, 1/K))$, represents the transform domain resampling effect which is the ratio of the transform-domain coefficients of the channel feature, $\mathbf{z}$, and that of its resampled version, $\mathcal{P}(\mathbf{z}, 1/K)$.

Let $a_0 = 1/a_0'$ and assume the term $\hat{T}(\mathbf{z})/\hat{T}_{1/K}(\mathcal{P}(\mathbf{z}, 1/K))$ can be represented by (3.5) and (3.20), in case of 2DDFT and 2DDCT, respectively. Then, the transform-domain coefficients of the original resolution, $\hat{\mathbf{Z}}$, can be approximated by using the transform-domain coefficients at a lower resolution, $\hat{\mathbf{Z}}_{1/K}$, as

$$\hat{\mathbf{Z}} \approx \alpha(K)\hat{\mathbf{Z}}_{1/K} \tag{3.27}$$

where

$$\alpha(K) = \begin{cases} a_0 K^{2-\lambda}, & \text{for 2DDFT} \\ a_0 K^{1-\lambda}, & \text{for 2DDCT} \end{cases} \tag{3.28}$$

In order to improve the approximation accuracy of expression in (3.27), we introduce

an additive correction term a_1, such that α is of the form

$$\alpha(K) = \begin{cases} a_0 K^{2-\lambda} + a_1, & \text{for 2DDFT} \quad (3.29a) \\ a_0 K^{1-\lambda} + a_1, & \text{for 2DDCT} \quad (3.29b) \end{cases}$$

The constants a_0, a_1, and λ are computed empirically in the training mode for the 2DHOG channel. The usefulness of $\alpha(K)$ given by (3.29) lies in the fact that the features extracted from a lower resolution test image can be utilized to approximate the features of the test image extracted at a higher resolution by multiplying the former by $\alpha(K)$, which is a function of the downsampling factor, K, and the type of transform.

Estimation of $a_0, a_1,$ ***and*** λ

Given a training set of N_t images, the parameters $a_0, a_1,$ and λ for the 2DHOG channel can be estimated as follows. First, at each value of the downsampling factor, $K = 1, 2, 4, ...,$ the multiplicative factor of the i^{th} image sample, $\hat{\alpha}^i(K)$, is obtained as the factor that minimizes the mean square error (MSE) as

$$\min_{\hat{\alpha}^i(K)} \frac{1}{N_y N_x c^2 \beta} \sum_{l,j,k,u,v} (\hat{\mathbf{Z}}^{i,j,k,l}[u,v] - \hat{\alpha}^i(K) \hat{\mathbf{Z}}^{i,j,k,l}_{1/K}[u,v])^2 \quad (3.30)$$

where $i = 1, ..., N_t$, $0 \leq u, v \leq c-1$, u and v are the frequency indices of the $(j,k)^{th}$ block, $1 \leq j \leq N_y$, $1 \leq k \leq N_x$, and $l = 1, 2, ..., \beta$. Then, the average value of the estimated multiplicative factor $\hat{\alpha}(K)$ is obtained as $\hat{\alpha}(K) = (1/N_t) \sum_{i=1}^{N_t} \hat{\alpha}^i(K)$. Finally, the values of the estimated multiplicative factor $\hat{\alpha}(K)$ are used to obtain the model parameters, a_0, a_1, and λ, of $\alpha(K)$ by using the least squares curve fitting. In Section 3.4.1, we compute empirically the values of $a_0, a_1,$ and λ.

3.3 Scheme for Vehicle Detection

In this section, we propose a new vehicle detection scheme by using the results of the previous section concerning TD2DHOG features so as to employ a single classifier trained on vehicles of high resolution in order to detect vehicles of the same or lower resolution, instead of training multiple resolution-specific classifiers, as in [34, 82]. In order to detect vehicles of different resolutions in a given test image, an image pyramid of depth one octave is constructed, and TD2DHOG features are extracted at each scale from the image pyramid with blocks of different sizes. We now present our methods for training and testing of the proposed vehicle detection scheme.

3.3.1 Training Mode

In order to take advantage of the fact that the transform-domain coefficients of the original resolution can be approximated by using the transform-domain coefficients at a lower resolution as given by (3.27), the training data is upsampled by a factor of R, R being an integer power of 2. Even though upsampling of the training data will cause an increase in the training cost, it has been observed from our experiments that training a classifier on TD2DHOG features obtained at a high resolution of images offers a detection accuracy higher than that achieved by the same classifier when trained on TD2DHOG features extracted from the same training set at a lower resolution. This is because of the fact that in the testing mode, going from a higher resolution to a lower resolution results in a smaller approximation error for TD2DHOG features than when going the other way around.

Figure 3.5 (a) shows the training scheme for the proposed vehicle detector, where the training data is upsampled by a factor R in both the x and y directions. Let the set of the training data upsampled by R be denoted as $\mathbf{I}_R = \{\mathbf{I}_{i,R}, i = 1, 2, ..., N_t\}$, where N_t denotes the number of training image samples. Then, the size of the i^{th} training image sample is $(RM_1 \times RM_2)$. Assume the 2DHOG features of the l^{th} layer,

$\mathbf{h}^l_{i,R}, (i = 1, 2, ..., N_t$ and $l = 1, 2, ..., \beta)$, are extracted by using the same cell size for all the resolutions $(\eta_1 \times \eta_2)$, then the size of the l^{th} 2DHOG layer of the i^{th} training image sample is $R\tilde{M}_1 \times R\tilde{M}_2$, i.e., increased by the same factor R. Similarly, the block size used to compute the corresponding TD2DHOG features is increased by the same factor R, i.e., $b_R = Rb_0$. We call b_0 as the *base block size*, which is defined as the block size at $R = 1$. Let $\hat{\mathbf{H}}^l_{i,R}, i = 1, 2, ..., N_t$, denote the TD2DHOG features of the l^{th} layer, where the size of $\hat{\mathbf{H}}^l_{i,R}$ is $(\hat{M}_1 \times \hat{M}_2)$. It is important to note that, in the training phase we do not multiply TD2DHOG features by the multiplicative factor $\alpha(K)$, and we use the value of $\alpha(K)$ computed from (3.29) in the detection phase.

After the extraction of the TD2DHOG features, 2DPCA [94] is employed on each layer in order to maintain the relation between the neighboring blocks. Let the training data consist of N_{pos} and N_{neg} training image samples, corresponding to the positive and negative classes, respectively. The training data can be denoted as $\{(\hat{\mathbf{H}}^l_{i,R}, y_i), i = 1, 2, ..., N_t\}, l = 1, 2, ..., \beta$, where $y_i \in \{+1, -1\}$ refers to the class label for the i^{th} image sample. The covariance matrix, of size $(\hat{M}_2 \times \hat{M}_2)$, is first obtained for the TD2DHOG features of the l^{th} layer as

$$\mathbf{Cov}^l = \frac{1}{N_t} \sum_{i=1}^{N_t} (\hat{\mathbf{H}}^l_{i,R} - \bar{\mathbf{H}}^l_R)^\top (\hat{\mathbf{H}}^l_{i,R} - \bar{\mathbf{H}}^l_R) \tag{3.31}$$

where

$$\bar{\mathbf{H}}^l_R = \frac{1}{N_t} \sum_{i=1}^{N_t} \hat{\mathbf{H}}^l_{i,R} \tag{3.32}$$

Note that $\mathbf{Cov}^l$ is a nonnegative definite matrix. Next, we obtain the r_l eigenvectors of $\mathbf{Cov}^l$ that correspond to the r_l dominant eigenvalues. The number of eigenvectors, r_l, is chosen so that the sum of the magnitude of the retained eigenvalues represents at least 90% of the sum of the magnitude of all the eigenvalues. The eigenvectors are used to form the matrix $\mathbf{V}^l_R$ of size $(\hat{M}_2 \times r_l)$. Next, the TD2DHOG features of the l^{th} layer of the i^{th} training image sample are projected onto the constructed matrix $\mathbf{V}^l_R$

in order to obtain the matrix $\mathbf{Q}_{i,R}^{l} = \hat{\mathbf{H}}_{i,R}^{l}\mathbf{V}_{R}^{l}$ of size $(\hat{M}_1 \times r_l)$, and $\mathbf{Q}_{i,R}^{l}$ is vectorized[1] to obtain the corresponding feature vector $\mathbf{q}_{i,R}^{l}$ of size $(1 \times \hat{M}_1 r_l)$. Then, for the i^{th} training image sample, the feature vectors from different layers, $\mathbf{q}_{i,R}^{l}$, are concatenated to obtain the feature vector, $\mathbf{f}_{i,R}$, of size $(1 \times r)$, where $\mathbf{f}_{i,R} = [\mathbf{q}_{i,R}^{1}, ..., \mathbf{q}_{i,R}^{\beta}]$ for $i = 1, 2, ..., N_t$.

Let the set of training features obtained after applying 2DPCA be denoted as $\mathcal{F}_R = \{\mathbf{f}_{i,R}, i = 1, 2, ..., N_t\}$, and the set of the eigenvectors used to generate these features be denoted as $\mathcal{V}_R = \{\mathbf{V}_{R}^{l}, l = 1, 2, ..., \beta\}$. Then, we train a classifier, $\mathcal{T}_R$, for the upsampling factor R by using the corresponding features $\mathcal{F}_R$. We use one of the two state-of-the-art classifiers: a support vector machine with fast histogram intersection kernel (FIKSVM) [22, 95] or boosted decision tree classifier (BDTC) [96, 97].

3.3.2 Testing Mode

In the testing phase, we first obtain an image pyramid of depth of one octave from the given input test image. The test image at each scale of the image pyramid, is then scanned by using a number of detection windows of different sizes as $(\frac{RM_1}{K} \times \frac{RM_2}{K})$, where R is the upsampling factor at which the detector has been trained and $K = 1, 2, 4, ...,$ an integer power of 2. Figure 3.5 (b) shows the proposed vehicle detection scheme when applied to a test image by assuming $R = 2$ and $K = 1$ and 2. Now for each detection window, we obtain the TD2DHOG features for different layers, $\{\hat{\mathbf{H}}_{test}^{l}, l = 1, 2, ..., \beta\}$ by using a block size $b^{test} = \frac{b_R}{K}$; the size of each $\hat{\mathbf{H}}_{test}^{l}$ is $(\hat{M}_1 \times \hat{M}_2)$. Then, the TD2DHOG features of each layer are multiplied by the multiplicative factor $\alpha(K)$ as

$$\tilde{\mathbf{H}}_{test}^{l} = \alpha(K)\hat{\mathbf{H}}_{test}^{l} \tag{3.33}$$

[1]The vectorization function is defined as Mat2Vec: $\mathbb{R}^{\mu\times\nu} \rightarrow \mathbb{R}^{\rho}$, where $\rho = \mu\nu$ is the dimension of the vector, and $(\mu \times \nu)$ is the order of the input matrix. The inverse of the vectorization function is defined as Vec2Mat: $\mathbb{R}^{\rho} \rightarrow \mathbb{R}^{\mu\times\nu}$.

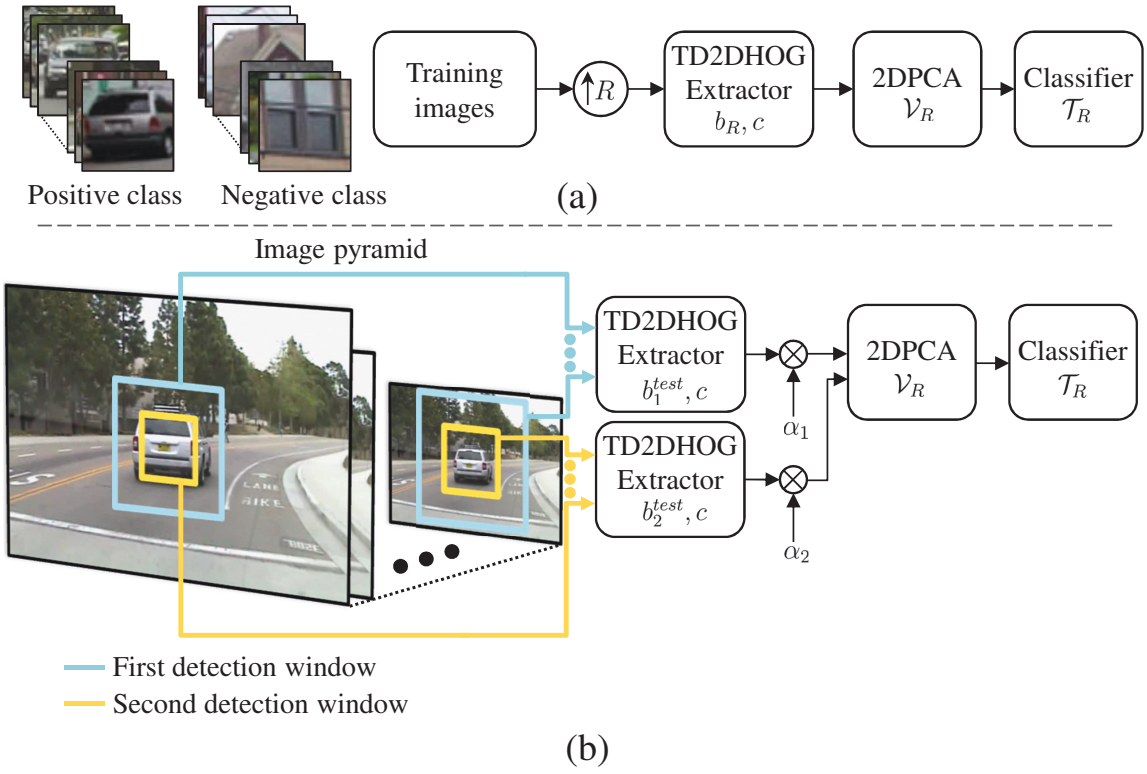


Figure 3.5: (a) The scheme for training the proposed vehicle detector with training images of size 64×64, where R is the upsampling factor in both the x and y directions. (b) Proposed vehicle detection scheme for a sample test image, where the different colors in the image pyramid represent different scanning window sizes (here we have used only two window sizes, 128×128 and 64×64).

where $\tilde{\mathbf{H}}^l_{test}$ is of size $(\hat{M}_1 \times \hat{M}_2)$, and $\alpha(K)$ is given by (3.29), which allows the TD2DHOG features obtained from a low resolution detection window to approximate the TD2DHOG features obtained at a higher resolution, indicating an approximate invariance of the TD2DHOG features within a multiplicative factor, when the image resolution is changed. Next, the TD2DHOG features of the l^{th} layer, $\tilde{\mathbf{H}}^l_{test}$, is projected onto the corresponding matrix $\mathbf{V}^l_R$ in order to obtain the matrix $\mathbf{Q}^l_{test} = \tilde{\mathbf{H}}^l_{test}\mathbf{V}^l_R$ of size $(\hat{M}_1 \times r_l)$. Then, $\mathbf{Q}^l_{test}$ is vectorized to obtain the corresponding feature vector $\mathbf{q}^l_{test}$ of size $(1 \times \hat{M}_1 r_l)$. This is followed by concatenating the features, $\mathbf{q}^l_{test}$, for different layers to obtain the feature vector, $\mathbf{f}_{test}$, of size $(1 \times r)$, where $\mathbf{f}_{test} = [\mathbf{q}^1_{test}, ..., \mathbf{q}^\beta_{test}]$.

Now, the trained classifier $\mathcal{T}_R$, namely, FIKSVM [22, 95] or BDTC [96, 97], is used to provide for each feature vector $\mathbf{f}_{test}$ a detection score corresponding to the input detection window. Finally, similar to [22], a non-maximum suppression technique is used to combine several overlapped detections for the same object. This avoids detecting the same vehicle more than once, and allows detecting vehicles with different aspect ratios.

Figure 3.6 (a) illustrates the scanning scheme for the proposed vehicle detector in the case of $R = 2$, and $K = 1$ and 2. Hence, in this example, the test image at each scale of the image pyramid is scanned by using two detection windows of sizes $(2M_1 \times 2M_2)$ and $(M_1 \times M_2)$. The proposed vehicle detector requires training a single classifier at the highest detection window size, namely, $(2M_1 \times 2M_2)$. The methods in [34, 82] use a similar scanning strategy; however, they require constructing a classifier pyramid in order to classify detection windows of different sizes. It is to be noted that the scanning scheme used in several state-of-the-art object detectors [6, 22, 23] requires the extraction of features at each scale of an image pyramid of depth often more than one octave, even though the scheme employs one detection window and a single classifier. Figure 3.6 (b) shows an example of this scanning scheme, when the image pyramid is of depth two octaves. The proposed vehicle detection scheme

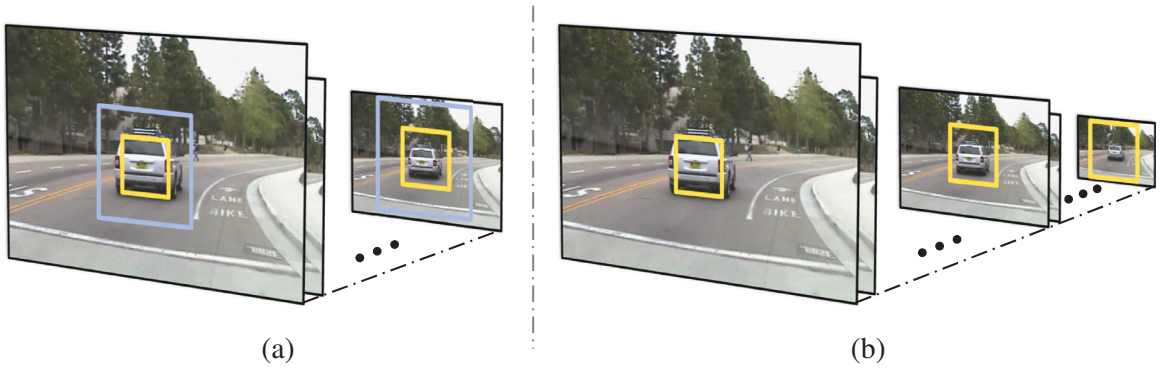


Figure 3.6: (a) An illustration of the proposed scheme for scanning an image pyramid of depth one octave with two detection windows and a single classifier. (b) An illustration of the scheme for scanning an image pyramid of depth two octaves with one detection window and a single classifier.

reduces the cost of training a classifier pyramid, as a single classifier trained on images of a given resolution can be used to detect vehicles of the same or lower resolutions. In addition, it reduces the storage requirements that are associated with training multiple resolution-specific classifiers.

3.4 Experimental Results

We first carry out a number of experiments to validate, as mentioned in Section 3.2, the model for the multiplicative factor $\alpha(K)$ using the *UIUC car detection dataset* [83]. Then, we study the performance of the proposed algorithm for vehicle detection in images using the *UIUC car detection dataset* [83], the *USC multi-view car detection dataset* [28], the *LISA 2010 dataset* [84] and the *HRI roadway dataset* [98]. We also compare the performance of our algorithm with that of some of the existing methods.

The *UIUC car detection dataset* [83] consists of 1050 training images of size 40×100 divided into a set of 550 car images with side views, and a set of 500 other images, none of which is the image of a car with a side view. In order to facilitate the computation of the TD2DHOG features, the training images in this dataset are cropped by removing pixels from the first and last four rows and from the first and last two columns in order to reduce the size of each image from 40×100 to 32×96. The testing images in this dataset consist of 108 multi-scale images. The dataset consists of partially occluded cars, objects with low contrast, as well as highly textured background. Since the dataset includes a balanced number of positive and negative training images, the FIKSVM [95] is used as the baseline classifier for the proposed detector.

The *USC multi-view car detection dataset* [28] consists of cars with several views. The training data consists of 2462 positive training images of size 64×128, while the testing data consists of 196 images containing 410 cars of different sizes and views. In order to complete the training dataset, we collect 9512 negative training image

samples from the *CBCL street scenes dataset* [99]. Since the USC dataset consists of cars with different views, BDTC [96, 97] is chosen as the baseline classifier.

The *LISA 2010 dataset* [84] consists of test sequences of size 480×704 for rear view vehicles of different sizes, and this dataset has been captured under several illumination conditions. The first sequence (1600 frames) is taken on a high-density highway during a sunny day (H-dense), which includes vehicles in partial occlusions, heavy shadows, and some background structures are confused with the positive class, while the second (300 frames) on a medium-density highway on a sunny day (H-medium), where this sequence includes challenges similar to H-dense but at a lower density. The dataset does not include training data; therefore, we collect training images of size 64×64 from other datasets as follows: (1) 9013 images of vehicles in rear/front views from *KITTI dataset* [100], and *USC multi-view car detection dataset* [28], and (2) 8415 negative image samples from *CBCL street scenes dataset* [99]. As in [84], we collect a number of hard negative image samples from the test sequences (229 image samples from H-medium, and 806 image samples from H-dense). Due to the large number of training samples and the wide variation in the background structures, BDTC [96, 97] is used as the baseline classifier on this dataset.

The *HRI roadway dataset* [98] consists of five test sequences of size 600×800 for vehicles on urban and highway areas. This dataset has been captured under several challenging weather and lighting conditions. Sequence I (908 frames) has been captured during a cloudy day, while Sequence II (917 frames) has been captured during a sunny day. Sequences III (611 frames), IV (411 frames) and V (830 frames) have been captured during a heavy rainy day, a dry midnight, and afternoon after a heavy snow, respectively. Since the HRI dataset does not have its own training set, in order to test the proposed scheme on a sequence of this dataset, the classifier in the proposed scheme is trained by employing the training set used in the case of *LISA 2010 dataset* along with the hard negative samples collected from the first 100 frames of this sequence of the HRI dataset.

3.4.1 Validation for the Model of $\alpha(K)$

We now validate the model of $\alpha(K)$ given by (3.29) by making use of the block diagram of Figure 3.4 and the scheme introduced in Section 3.2.2 for estimating the channel parameters a_0, a_1 and λ. For this purpose, we first consider the UIUC car detection dataset [83] and choose $N_t = 550$ car images. Since we do not have access to high resolution versions of these car images, they are upsampled by a factor $R = 8$. Now, we give the procedure to estimate the value of $\alpha(K)$ for the 2DHOG features in the 2DDFT domain. We first obtain the 2DHOG features of an upsampled image[1], $\mathbf{I}_u$, using the steps outlined in Section 2.1, assuming $\eta_1 = \eta_2 = 4$, and $\beta = 5, 7$ or 9. We then apply 2DDFT on block-partitioned 2DHOG features given by (3.21) for each of the layers, assuming the block size to be $b = Rb_0 = 8b_0$, $b_0 \in \{4, 8, 16\}$. This is followed by a truncation operation retaining the $(c \times c)$ low frequency coefficients, where $c = 4$, to obtain the 2DHOG features in the 2DDFT domain. Then, the whole operation is repeated after downsampling $\mathbf{I}_u$ by a factor K, $K = 1, 2, 4$, and 8, but with a block size of b/K. As explained in Section 3.2.2, the multiplicative factor of the i^{th} image sample, $\hat{\alpha}^i(K)$, is obtained as the factor that minimizes the mean square error (MSE) given by (3.30). Then, the four values of the estimated multiplicative factor $\hat{\alpha}(K)$, $K = 1, 2, 4$, and 8, are used to obtain the model parameters, a_0, a_1, and λ, of $\alpha(K)$ by using the least squares curve fitting[2]. The above procedure is repeated to find the model parameters, a_0, a_1, and λ, of $\alpha(K)$ for the 2DHOG features in the 2DDCT domain.

Table 3.1 summarizes the values of the parameters, a_0, a_1, and λ, for the above two cases for block size $b_0 = 4, 8, 16$ along with the corresponding mean square errors, when the number of layers, β, is 5, 7, or 9. It is seen from this table that irrespective of the transform used, the errors are insignificant. Figure 3.7 shows the plots of $\alpha(K)$

[1]The toolbox [97] has been used to calculate the 2DHOG.

[2]The MATLAB function `lsqcurvefit` is used, `http://www.mathworks.com/help/optim/ug/lsqcurvefit.html`

for the 2DHOG features for $\beta = 7$. It is seen from these plots that the proposed model is not sensitive to the block size b_0. It has been observed that $\alpha(K)$ is insensitive to b_0 for the other values of β also.

Similar studies have been conducted using $N_t = 1000$ positive training images from the USC multi-view car detection dataset, and $N_t = 1000$ positive training images, collected as mentioned earlier in this section, for the LISA 2010 dataset. It has been found that for both these datasets, $\alpha(K)$ is insensitive to b_0 irrespective of whether $\beta = 5, 7$ or 9.

It is to be noted that had we used the same model for $\alpha(K)$ as given by (3.29) also for the case of grayscale (GS) channel in the 2DDFT and 2DDCT domains and repeated the above procedures, we would obtain the values of a_0, a_1 and λ. These values for the 2DDFT and 2DDCT domains are also included in Table 3.1 using the UIUC car detection dataset. It is seen from this table that for the case of the grayscale channel, $\lambda \approx 0, a_0 \approx 1$ and $a_1 \approx 0$, and thus,

$$\alpha(K) \approx \begin{cases} K^2, & \text{for 2DDFT} \\ K, & \text{for 2DDCT} \end{cases} \tag{3.34}$$

Equation (3.34) has been found to be equally true in the case of the other two datasets, namely, the USC multi-view car detection dataset and the LISA 2010 dataset. It is seen that the two expressions on the right side of (3.34) are the same as that given by (3.5) and (3.20), respectively, when $K_1 = K_2 = K$. Thus, the proposed model for $\alpha(K)$ given by (3.29) for the TD2DHOG features is also valid for the grayscale images in the transform domain. These results show the versatility of the model for $\alpha(K)$ in representing channels other than the 2DHOG channel.

Table 3.1: The estimated channel parameters for grayscale image (GS) and 2DHOG features, where $b_0 = 4, 8,$ or 16, and MSE refers to the mean square error of the curve fitting

		GS		2DHOG					
		2DDFT	2DDCT	2DDFT			2DDCT		
				$\beta = 5$	$\beta = 7$	$\beta = 9$	$\beta = 5$	$\beta = 7$	$\beta = 9$
$b_0 = 4$	λ	0.00635	−0.00436	0.51538	0.53305	0.54992	−0.79613	−0.85311	−0.87422
	a_0	1.00846	0.99210	0.51819	0.52523	0.53464	0.01179	0.00950	0.00834
	a_1	−0.01189	0.00850	0.52819	0.52095	0.51050	0.98753	0.99027	0.99170
	MSE	0.00001	0.00000	0.00251	0.00252	0.00243	0.00000	0.00000	0.00000
$b_0 = 8$	λ	0.00060	−0.00085	0.51906	0.53351	0.54607	−0.81072	−0.87074	−0.89513
	a_0	1.00055	0.99831	0.51119	0.51558	0.52167	0.01048	0.00846	0.00751
	a_1	−0.00067	0.00183	0.53831	0.53411	0.52742	0.98901	0.99134	0.99245
	MSE	0.00000	0.00000	0.00286	0.00291	0.00286	0.00000	0.00000	0.00000
$b_0 = 16$	λ	0.00036	0.00011	0.52324	0.53168	0.53758	−0.79483	−0.83676	−0.85726
	a_0	1.00043	1.00014	0.51639	0.51853	0.52071	0.01107	0.00959	0.00883
	a_1	−0.00057	−0.00014	0.53153	0.52958	0.52731	0.98824	0.98991	0.99077
	MSE	0.00000	0.00000	0.00269	0.00273	0.00273	0.00000	0.00000	0.00000

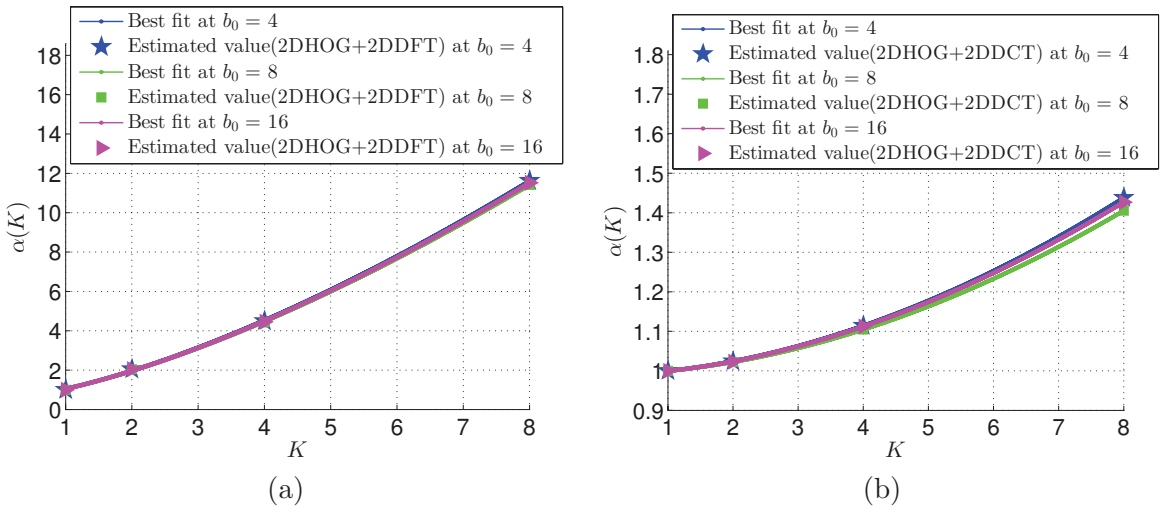


Figure 3.7: The multiplicative factor $\alpha(K)$ for $K = 1, 2, 4, 8$, where (a) and (b) represent the case of the 2DHOG features in the 2DDFT and 2DDCT domains, respectively.

3.4.2 Vehicle Detection using TD2DHOG Features

In this section, we study the detection performance of the proposed scheme using the datasets mentioned earlier. Further, the detection performance of the proposed technique is compared with that of several state-of-the-art techniques. The 2DHOG is obtained assuming $\eta_1 = \eta_2 = 4$ from which the TD2DHOG features are obtained. In case of using a single classifier, the TD2DHOG features multiplied by the factor $\alpha(K)$ given by (3.29) are used, where the classifier is trained on TD2DHOG features obtained from training images upsampled by a factor R and used to classify images in the detection windows of the same or lower resolutions. We refer to this scheme using a single classifier (SC) as TD2DHOG-SC. Also, we consider the case of using multiple classifiers trained on TD2DHOG features at different values of R in order to classify images in the detection windows at the same resolution at which the classifier has been trained. We refer to this scheme using a classifier pyramid (CP) as TD2DHOG-CP. Unless specified otherwise, each octave of an image pyramid is considered to have 12 scales. Each scale is scanned by shifting the detection window(s) by $8R$ pixels in each of the x and y directions.

a) UIUC Car Detection Dataset

On this dataset the equal error rate (EER) is used for evaluation, EER being the detection rate at the point of equal precision and recall; we use the methodology given in [83] to calculate the precision and recall.

Choice of the Transform: In this experiment, we evaluate the detection performance of the proposed TD2DHOG-SC by using 2DDFT or 2DDCT. The TD2DHOG features are obtained assuming $b^{train} = Rb_0$, $R = 2$, $b_0 = 4$, $c = 4$, $b^{test} = 4, 8$ and $\beta = 5, 6, ..., 11$. Figure 3.8 shows that DCT2DHOG-SC exhibits a better performance irrespective of β. Similar results have been obtained for other datasets. In view of this, we will henceforth consider only DCT2DHOG features in all the experiments.

Choice of b_0, c, and β: We now study the performance of the proposed DCT2DHOG-SC for different values of b_0, c and β, in order to make an appropriate choice for these parameters. Figure 3.9 shows the EER values of the proposed DCT2DHOG-SC for $b_0 = 4, c = 2$ or 4; $b_0 = 8, c = 2, 4$ or 8 with $\beta = 5, 6, ..., 11$ and $b^{test} = b_0$ and $2b_0$. It is observed from this figure that the highest EER value is achieved at three different parameter settings, $b_0 = 4, c = 4, \beta = 7, b_0 = 4, c = 4, \beta = 9$, and $b_0 = 8, c = 8, \beta = 7$. We choose the parameter setting $b_0 = 4, c = 4, \beta = 7$, since it retains the lowest number of eigenvectors compared to that of the other two parameter settings and thus it offers the lowest detection complexity. It has also been observed that in the case of DCT2DHOG-CP, the parameter setting $b_0 = 4, c = 4$ and $\beta = 7$ also provides the best EER value.

Performance Evaluation: We first consider the case of the DCT2DHOG-SC scheme. In this case, the single classifier trained at $R = 2$ is used to classify the test images in detection windows with the same or lower resolutions (by making use of $\alpha(K)$, which is obtained using Table 3.1 and (3.29b)), where the test block sizes used are $b^{test} = 8$ and 4.

Now, we consider the case of DCT2DHOG-CP. In this case, we construct a classifier pyramid trained at $R = 1$ and 2. These two classifiers are used to classify the test images in detection windows of the corresponding two resolutions, where $b^{test} = 4$ and 8, respectively.

For each of the above cases, EER values are computed and are given in Table 3.2. The EER values corresponding to several state-of-the-art schemes, namely, Gabor filter-based technique [101], implicit shape model [12], bag of words with spatial pyramid kernel [102], discriminative parts with Hough forest [103], contour cue-based technique [104], HOG-based technique of [28], aggregated channel feature (ACF) and ACF-Exact [33], and Multi-resolution 2DHOG [95] are also included in Table 3.2. It is seen from this table that the performance of either of the two proposed schemes is better than that of the others in [12, 28, 33, 95, 101–104].

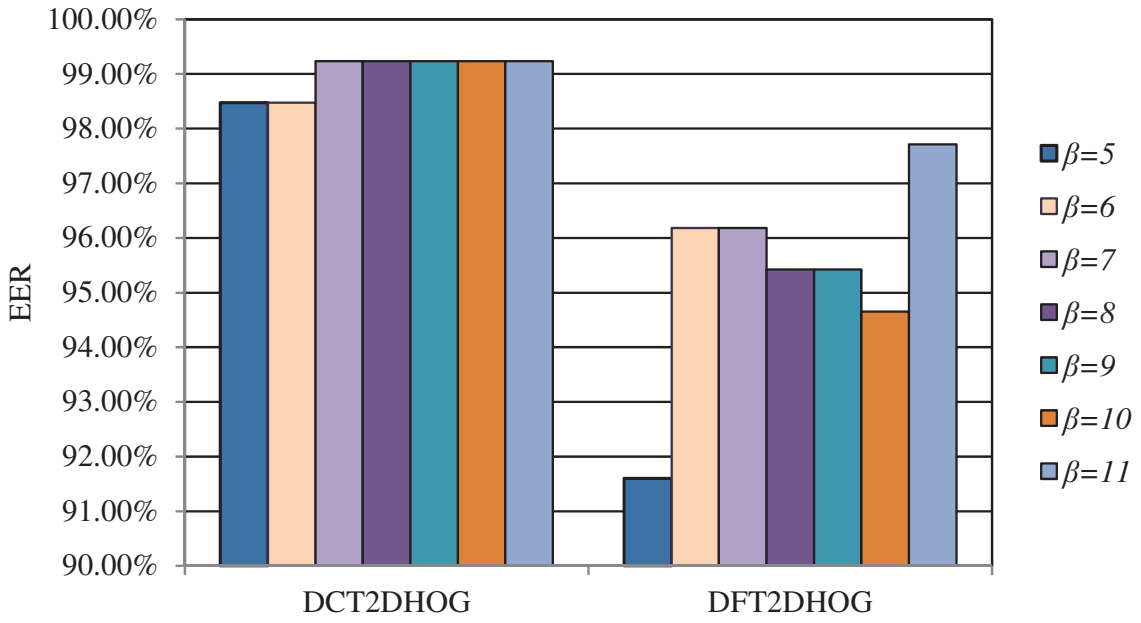


Figure 3.8: Comparing the EER values of the DFT2DHOG-SC and DCT2DHOG-SC on UIUC dataset.

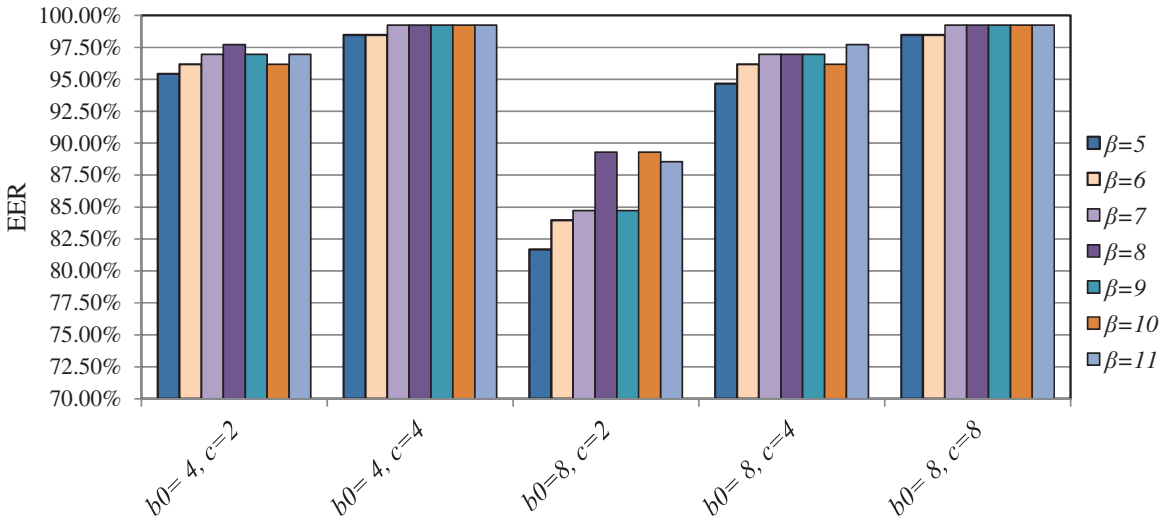


Figure 3.9: EER value of the proposed scheme DCT2DHOG-SC at $c = 2, 4$ or 8 obtained on the UIUC dataset, where $\beta = 5, 6, ...,$ or 11, and the base block size $b_0 = 4$ or 8.

Table 3.2: Equal Error Rate on UIUC car detection dataset

Method	EER
DCT2DHOG-SC($b_0 = 4$, $c = 4$, $\beta = 7$)	**99.28%**
DCT2DHOG-CP($b_0 = 4$, $c = 4$, $\beta = 7$)	**99.28%**
Lampert *et al.* [102]	<u>98.60%</u>
Gall and Lempitsky [103]	<u>98.60%</u>
Wu *et al.* [104]	97.80%
Dollár *et al.* [33] (ACF - Exact)*	97.12%
Dollár *et al.* [33] (ACF)*	95.68%
Maji *et al.* [95]*	95.68%
Kuo and Nevatia [28]	95.00%
Leibe *et al.* [12]	95.00%
Mutch and Lowe [101]	90.60%

Note: $*$ denotes the results obtained by utilizing the code provided by the authors of the paper. The best and the second best results are shown in boldface and underscored, respectively.

b) USC Multi-view Car Detection Dataset

For this dataset, as in [28], the PASCAL visual object classes (VOC) criterion [105, 106] is used for the evaluation purpose with an overlap threshold of 0.5. To compare the performance of our method to that of some recent schemes, the average precision (AP) is used as an evaluation metric. In this dataset, the training images are upsampled by a factor of $R = 1$ and 2 in the case of using DCT2DHOG-CP, and by a factor of $R = 2$ in the case of using DCT2DHOG-SC. The performance of the proposed DCT2DHOG-SC scheme using this dataset is studied for $b_0 = 4, c = 4$; $b_0 = 8, c = 4$ or 8; and $\beta = 5, 7$ or 9, and $b^{test} = b_0$ and $2b_0$. It is observed that the highest AP value is achieved at two parameter settings, $b_0 = 8$, $c = 4$, $\beta = 9$ and $b_0 = 8$, $c = 8$, $\beta = 9$. We choose the parameter setting $b_0 = 8$, $c = 4$ and $\beta = 9$, since

Table 3.3: Average Precision on USC Multi-view Car Dataset

Method	AP
DCT2DHOG-SC($b_0 = 8$, $c = 4$, $\beta = 9$)	**90.44%**
DCT2DHOG-CP($b_0 = 8$, $c = 4$, $\beta = 9$)	<u>89.92%</u>
ACF - Exact [33]*	89.31%
ACF [33]*	89.64%
Multi-resolution 2DHOG [95] - BDTC*	89.38%
Kuo and Nevatia [28]	85.61%
Wu and Nevatia [107]	52.55%

Note: $*$ denotes the results obtained by utilizing the code provided by the authors of the paper. The best and the second best results are shown in boldface and underscored, respectively.

it retains a lower number of 2DDCT coefficients than that of the other parameter setting, and thus it provides a lower detection complexity. Therefore, this parameter setting is chosen for both the DCT2DHOG-SC and DCT2DHOG-CP schemes.

Figure 3.10 shows sample qualitative results for the proposed scheme on this dataset. It shows that the proposed scheme can detect cars in different views and resolutions. Table 3.3 shows that the performance of the proposed technique is better than that of the method in [28] which uses HOG with Gentle AdaBoost, and that of the method in [107] which is based on using Edgelet feature with cluster boosted tree classifier, where the latter is evaluated using [28]. Further, the performance of the proposed method is slightly better than that of the implementations of the methods in [33], or that of the multi-resolution 2DHOG features presented in [95] when used with BDTC. The proposed scheme achieves AP values of 90.44% in the case of DCT2DHOG-SC, and 89.92% in the case of DCT2DHOG-CP. Thus, DCT2DHOG with a single classifier exhibits a high detection performance, while requiring the training of only a single classifier, instead of multiple classifiers for each resolution.

Figure 3.10: Sample results for the proposed scheme when applied on USC multi-view car dataset, where colors represent: (blue) true positive, and (red) false positive.

c) LISA 2010

In this dataset, the same evaluation metrics presented in [84] are used, namely, true positive rate (TPR) or recall, false detection rate (FDR) or (1 - precision), average false positive per frame (AFP/F), average false positive per object (AFP/O), and average true positive per frame (ATP/F). These metrics are computed at the point of equal precision and recall. True positive detections are computed by using the PASCAL VOC criterion [105, 106] with an overlap threshold of 0.5.

On both the H-dense and H-medium sequences, the single classifier trained at $R = 2$ is used in the case of DCT2DHOG-SC and two classifiers trained at $R = 1$ and 2 are used in the case of DCT2DHOG-CP. As in our experiments on USC multi-view car detection dataset, the parameter setting chosen for both the DCT2DHOG-SC and DCT2DHOG-CP schemes on LISA 2010 dataset is $b_0 = 8, c = 4, \beta = 9$, and $b^{test} = 8$ and 16, since, these two datasets contain similar environmental conditions and the same type of classifier, namely, BDTC, is used in the detection process.

Table 3.4 gives the detection performance of the proposed method, from which it is clear that the performance of DCT2DHOG using a single classifier is almost as good as that of using classifier pyramid. Table 3.4 also lists the performance of some of the other methods, namely, the Haar-like features-based technique presented in [84], ACF and ACF-Exact [33], and multi-resolution 2DHOG [95]. From this table, it can be seen that the proposed scheme on H-medium sequence provides a performance better than that of the schemes of [33, 84, 95], while for the H-dense sequence, our scheme provides 92.67% TPR at 6.03% FDR, which is better than that of the methods in [33, 95]. The proposed method and the methods in [33, 95] are trained with hard negative samples collected from the *CBCL street scenes dataset* [99], while the method in [84] is trained on private data from sunny highway environment. The detection performance of the proposed scheme can be improved by using an online learning technique to incorporate the false positive samples in the learning process.

Figure 3.11 (a) shows sample qualitative results for the proposed scheme when

Table 3.4: The performance for the proposed scheme on LISA dataset

	Method	TPR	FDR	AFP/F	ATP/F	AFP/O
H-dense	DCT2DHOG-SC	<u>92.67%</u>	**6.03%**	**0.26**	<u>4.06</u>	**0.06**
	DCT2DHOG-CP	<u>92.67%</u>	**6.03%**	**0.26**	<u>4.06</u>	**0.06**
	Sivaraman and Trivedi [84]	**93.50%**	<u>7.10%</u>	<u>0.32</u>	**4.2**	<u>0.07</u>
	ACF - Exact [33]*	87.43%	12.54%	0.55	3.83	0.13
	ACF [33]*	86.75%	13.23%	0.58	3.8	0.13
	Multi-resolution 2DHOG [95]*	73.24%	26.76%	1.17	3.21	0.27
H-medium	DCT2DHOG-SC	98.11%	<u>1.89%</u>	<u>0.06</u>	2.94	**0.02**
	DCT2DHOG-CP	<u>98.22%</u>	**1.78%**	**0.05**	<u>2.95</u>	**0.02**
	Sivaraman and Trivedi [84]	**98.80%**	10.30%	0.37	**3.18**	0.11
	ACF - Exact [33]*	93.11%	6.89%	0.21	2.79	0.07
	ACF [33]*	94.33%	5.67%	0.17	2.83	<u>0.06</u>
	Multi-resolution 2DHOG [95]*	77.44%	19.70%	0.57	2.32	0.19

Note: $*$ denotes the results obtained by utilizing the code provided by the authors of the paper. The best and the second best results on each dataset are shown in boldface and underscored, respectively.

applied on the H-dense sequence. As mentioned earlier, this sequence contains heavy shadows, vehicles in partial occlusions and some background structures are confused with the positive class. The proposed scheme can detect correctly 92.67% from the vehicles under these challenging conditions. Figure 3.11 (b) shows the corresponding results for the H-medium sequence, which includes challenges similar to that of the H-dense sequence but at a lower density. It is clear that the proposed technique can detect vehicles of various resolutions, under different illumination and background conditions.

d) HRI Roadway Dataset

For this dataset, the evaluation metrics presented in Section (3.4.2.c) are used. As in our experiments on the USC multi-view car detection dataset and LISA 2010 dataset, the single classifier trained at $R = 2$ is used in the case of DCT2DHOG-SC and two classifiers trained at $R = 1$ and 2 are used in the case of DCT2DHOG-CP for all the five test sequences of the HRI dataset. Also, the same parameter setting is chosen for both the DCT2DHOG-SC and DCT2DHOG-CP schemes, namely, $b_0 = 8$; $c = 4$; $\beta = 9$, and $b_{test} = 8$ and 16. The choice of these parameters is made since these three datasets contain similar challenging conditions and the type of the classifier used is the same, namely, BDTC.

Table 3.5 shows the detection performance of DCT2DHOG-SC, DCT2DHOG-CP, and other state-of-the-art techniques, namely, ACF and ACF-Exact [33], and multi-resolution 2DHOG [95]. From this table, it can be seen that for sequences I, II and IV either of the DCT2DHOG-SC and DCT2DHOG-CP schemes provides TPR values better than that in case of the schemes in [33, 95], whereas for the sequences III and V, the DCT2DHOG-SC scheme yields TPR values higher than that in case of DCT2DHOG-CP or when the schemes of [33] and [95] are used. Note that the sequences III and V have been captured in heavy rain and snowy conditions, respectively.

Figure 3.11: Sample qualitative results for the proposed method on LISA 2010 dataset, such that (a) Highway-dense sequence, (b) Highway-medium or sunny sequence: (blue) true positive, and (red) false positive.

Table 3.5: The performance for the proposed scheme on HRI dataset

	Method	TPR	FDR	AFP/F	ATP/F	AFP/O
Sequence I	DCT2DHOG-SC	**78.13%**	**21.88%**	**0.16**	0.56	**0.22**
	DCT2DHOG-CP	**78.13%**	**21.88%**	**0.16**	0.56	**0.22**
	ACF - Exact [33]*	68.29%	31.71%	0.48	**1.04**	0.32
	ACF [33]*	66.67%	33.33%	0.52	**1.04**	0.33
	Multi-resolution 2DHOG [95] - BDTC*	68.97%	31.03%	0.36	0.80	0.31
Sequence II	DCT2DHOG-SC	**67.86%**	**32.14%**	**0.20**	0.42	**0.32**
	DCT2DHOG-CP	**67.86%**	**32.14%**	**0.20**	0.42	**0.32**
	ACF - Exact [33]*	65.63%	34.38%	0.39	**0.75**	0.34
	ACF [33]*	60.61%	39.39%	0.45	0.69	0.39
	Multi-resolution 2DHOG [95] - BDTC*	53.85%	46.15%	0.29	0.34	0.46
Sequence III	DCT2DHOG-SC	**72.73%**	27.27%	0.30	0.80	0.27
	DCT2DHOG-CP	66.67%	33.33%	0.37	0.73	0.33
	ACF - Exact [33]*	66.67%	20.00%	**0.29**	1.18	**0.17**
	ACF [33]*	72.41%	**19.23%**	0.31	**1.31**	**0.17**
	Multi-resolution 2DHOG [95] - BDTC*	45.45%	54.55%	0.60	0.50	0.55
Sequence IV	DCT2DHOG-SC	73.33%	26.67%	0.20	0.55	0.27
	DCT2DHOG-CP	**80.00%**	**20.00%**	**0.15**	0.60	**0.20**
	ACF - Exact [33]*	63.16%	36.84%	0.50	**0.86**	0.37
	ACF [33]*	63.16%	36.84%	0.50	**0.86**	0.37
	Multi-resolution 2DHOG [95] - BDTC*	73.33%	26.67%	0.20	0.55	0.27
Sequence V	DCT2DHOG-SC	**66.67%**	33.33%	0.22	0.44	0.33
	DCT2DHOG-CP	62.16%	37.84%	**0.02**	0.03	0.38
	ACF - Exact [33]*	64.00%	**23.81%**	0.36	**1.14**	0.20
	ACF [33]*	61.54%	**23.81%**	0.33	1.07	**0.19**
	Multi-resolution 2DHOG [95] - BDTC*	51.85%	48.15%	0.32	0.34	0.48

Note: $*$ denotes the results obtained by utilizing the code provided by the authors of the paper. The best and the second best results on each dataset are shown in boldface and underscored, respectively.

e) Discussion

In this section, we present an evaluation of the proposed scheme in terms of the cost for the training and testing schemes. For a fair comparison, we use 2DPCA and FIKSVM or 2DPCA and BDTC as the main building blocks when 2DHOG or DCT2DHOG features are used. In the experiments that follow, the same values of η_1, η_2, b_0, c, and β that have been used to obtain the detection accuracy on the corresponding dataset are used. It should be noted that in practical situations, the choice of these parameters depends on the targeted vehicle view. In case the side view of the vehicles is of interest, the parameter settings recommended for obtaining DCT2DHOG features are $b_0 = 4, c = 4$, and $\beta = 7$ and FIKSVM can provide a fast and accurate classification scheme. In the case of detecting vehicles with different views, such as the situations that exist in urban and highway scenarios, the recommended parameter settings are $b_0 = 8, c = 4$, and $\beta = 9$ and BDTC is preferred, since it can be trained on a large number of samples and can capture large intra-class variations that exist within the positive class samples.

Training Cost: In this experiment, we compare the training cost of the proposed DCT2DHOG against that of 2DHOG at six different resolutions. Table 3.6 lists the overall training time[1] of the proposed DCT2DHOG at six resolutions along with that of 2DHOG. It is seen from this table that the training time for the proposed scheme is less than that of 2DHOG by at least 49.79% when a classifier pyramid is used, and by at least 74.33% when a single classifier trained at $R = 2$ is employed. Table 3.7 gives the storage requirement of the proposed scheme and that of the 2DHOG-based scheme for classifiers trained at the six different resolutions considered. It is seen from this table that the storage requirement for the proposed scheme is lower than that of 2DHOG-based scheme in case of the UIUC dataset by 64.18% when the size of the detection window is 64×192, whereas both these schemes achieve the same storage for the cases of USC and LISA 2010 datasets. Note that the FIKSVM classifier is used

[1]Using modern computer of 2.9GHz CPU, and 8G RAM

Table 3.6: Feature extraction and classifier training times (in seconds) for the proposed DCT2DHOG method and for the 2DHOG method

Dataset		UIUC		USC		LISA 2010	
$M_1 \times M_2$		32×96	64×192	64×128	128×256	64×64	128×128
DCT2DHOG	FET	**8.00**	**9.72**	**245.87**	**283.91**	**85.03**	**107.22**
	CTT	**6.75**	**5.71**	**14.32**	**13.49**	**7.63**	**7.76**
	TT	**14.75**	**15.43**	**260.19**	**297.40**	**92.66**	**114.98**
2DHOG	FET	8.53	11.76	604.70	2133.36	291.74	806.56
	CTT	7.36	32.46	54.14	170.76	52.01	141.11
	TT	15.89	44.22	658.84	2304.11	343.74	947.67
Reduction in TT (CP)		49.79%		81.18%		83.92%	
Reduction in TT (SC)		74.33%		89.96%		91.10%	

Note: FET: Time in seconds for feature extraction, CTT: Time in seconds for training a classifier, TT: Average training time in seconds, Reduction in TT (CP) and (SC) refer to the amount of reduction in TT of DCT2DHOG-CP method over 2DHOG method, and DCT2DHOG-SC method over 2DHOG method, respectively.

for the UIUC dataset and BDTC is used for the USC and LISA 2010 datasets. It is observed from Tables 3.6 and 3.7, in order to detect vehicles of different resolutions, the proposed DCT2DHOG-SC requires only a single classifier instead of multiple ones, resulting in a reduction in terms of the training cost by at least 44.63% and the storage requirement by at least 50.00% compared with that of DCT2DHOG-CP.

It is to be pointed out that the reduction in the training and storage costs is achieved by the proposed vehicle detector in comparison with that of the 2DHOG counterpart using a classifier pyramid with almost no loss in the detection accuracy.

Detection Time: Table 3.8 gives a comparison of the feature extraction time as well as the detection time (in seconds) of the proposed transform-domain based detector

Table 3.7: Storage requirements (in MByte) for the proposed DCT2DHOG method and for the 2DHOG methods

Dataset	UIUC		USC		LISA 2010	
$M_1 \times M_2$	32×96	64×192	64×128	128×256	64×64	128×128
DCT2DHOG	1.51	2.16	0.21	0.21	0.21	0.21
2DHOG	1.51	6.03	0.21	0.21	0.21	0.21
Reduction in storage (CP)	51.33%		0.00%		0.00%	
Reduction in storage (SC)	71.35%		50.00%		50.00%	

Note: Reduction in storage (CP) and (SC) refer to the amount of reduction in storage of DCT2DHOG-CP method over 2DHOG method, and DCT2DHOG-SC method over 2DHOG method, respectively.

(Method A) with that of the spatial-domain counterparts (Methods B and C) on the three vehicle detection datasets, UIUC [83], USC [28] and LISA 2010 [84]. We use test images of size 480×640. We assume that each octave of an image pyramid consists of 8 scales, and that each scale is scanned by shifting the detection window(s) by 16 pixels in each of the x and y directions. This generates 1398, 1141 and 1365 detection windows per frame for UIUC, USC and LISA 2010 datasets, respectively.

Method A in Table 3.8 corresponds to the proposed method, where the DCT2DHOG-2DPCA features are used to train a single classifier at $R = 2$. Further, two detection windows of different sizes are used to scan an image pyramid of depth one octave and the same classifier is used to classify DCT2DHOG-2DPCA features obtained from images within these detection windows after incorporating the multiplicative factor $\alpha(K)$ given by (3.29b).

Method B corresponds to the traditional method that uses a single classifier trained on features obtained in the spatial domain, namely, 2DHOG-2DPCA features, at $R = 1$. Further, it uses a single detection window to scan an image pyramid of depth two octaves. Then, the 2DHOG-2DPCA features obtained from an image

within a detection window are classified by the trained classifier.

Method C corresponds to a spatial domain method which uses 2DHOG-2DPCA features to train two classifiers at $R = 1$, and 2. Further, two detection windows of different sizes are used to scan an image pyramid of depth one octave. Then, the two classifiers trained at $R = 1$ and 2 are used to classify images within the detection windows of the same resolution at which the classifier is trained.

For the UIUC dataset, the first detection window is of size 32×96 and the second one of size 64×192. For this dataset, the range of vehicle size that can be detected by using the method A, B or C is 32×96 to 128×384. For USC and LISA 2010 datasets the corresponding window sizes are 64×128 and 128×256, and 64×64 and 128×128, respectively.

It is seen from Table 3.8 that the proposed transform-based method provides a minimum of 4.69% reduction in the feature extraction time and a minimum of 17.82% reduction in the detection time over that of the two spatial-domain methods B and C for the UIUC dataset and very much higher reductions for the other two datasets.

Finally, it is worth mentioning that the classification time of the proposed method represents on average about 65% of the total detection time. Thus, further gains in the detection speed could be achieved by reducing the classification time.

Table 3.8: Average feature extraction and detection time in seconds for Methods A, B and C applied to three datasets

Dataset		UIUC	USC	LISA 2010
Range of vehicle size		32×96 to 128×384	64×128 to 256×512	64×64 to 256×256
Number of detection windows per frame		1398	1141	1365
Method A	FET	**0.061**	**0.077**	**0.059**
	DT	**0.143**	**0.212**	**0.218**
Method B	FET	0.064	0.112	0.122
	DT	0.174	0.397	0.475
Method C	FET	0.073	0.130	0.137
	DT	0.301	0.376	0.375
Min. reduction in FET		4.69%	31.25%	51.64%
Min. reduction in DT		17.82%	43.62%	41.87%

Note: FET: feature extraction time in second, DT: detection time in second, Min. reduction in FET and DT refer to the minimum amount of reduction in FET and DT of Method A over those of Methods B and C.

3.5 Summary

In this chapter, we have introduced transform domain features of two-dimensional histogram of oriented gradients of images, referred to as TD2DHOG features [81, 82]. Then, we have studied the effect of image downsampling on the TD2DHOG features. It has been shown that the TD2DHOG features obtained from a high resolution image can be approximated by using the TD2DHOG features obtained from the image at a lower resolution by multiplying the latter by a factor that depends on the downsampling factor. A model for this multiplicative factor has been proposed and validated experimentally in the case of 2DDFT and 2DDCT domains. Next, a novel vehicle detection scheme using these TD2DHOG features has been proposed. It has been shown that the use of TD2DHOG features reduce the cost of training a classifier pyramid, since a single classifier can be used to detect vehicles of the same or lower resolution at which the classifier has been trained, instead of training multiple resolution-specific classifiers. Experimental results have shown that when the proposed TD2DHOG features are used with the multiplying factor and a single classifier for vehicle detection, it provides a detection accuracy similar to that obtained using these features with a classifier pyramid; however, the use of a single classifier has a significant advantage over the use of a classifier pyramid in that the former results in substantial savings in training and storage costs. In addition, the proposed method provides a detection accuracy that is similar or even better than that provided by the state-of-the-art techniques.

Chapter 4

Online Multi-Object Tracking via Robust Collaborative Model and Sample Selection

In this chapter, we develop a collaborative model for interaction between a number of single-object online trackers and a pre-trained object detector, and use it in proposing a novel online multi-object tracking (MOT) scheme [85, 86] that is robust to the false positives and missed detections. In Section 4.1, we present a general architecture for the proposed multi-object tracking scheme, consisting of a pre-trained object detector, a data association module and a number of single-object trackers. In Section 4.2 the proposed tracking scheme is presented. First, we introduce the particle filter which uses the proposed collaborative model. Next, the appearance model of the proposed tracker, which uses discriminative and generative appearance models, is presented. A new image sample selection scheme is then introduced to update each tracker by using relevant samples from its trajectory. Finally, a data association scheme that can handle partial occlusion is introduced. In Section 4.3, extensive experiments on benchmark datasets are conducted to evaluate the performance of the proposed multi-object tracking scheme and compare it with that of the state-of-the-art methods. Finally, a summary of the work presented in this chapter is provided in Section 4.4.

4.1 General Architecture of the Proposed Scheme

The proposed multi-object tracking scheme consists of three main components: a pre-trained object detector, a data association module and a number of single-object trackers. Figure 4.1 shows the block diagram of the proposed scheme, wherein only one single-object tracker is shown. The object detector is applied on every frame and supports the data association module with a set of detections $\mathcal{D}^t$ at time t. The object tracker adopts a hybrid motion model, and a particle filter with a collaborative model is used to estimate the target location. The appearance model consists of a sparsity-based discriminative classifier (SDC) with holistic features, a sparsity-based generative model (SGM) with local features, and a 2DPCA-based generative model (PGM) with holistic features. The SDC is used to compute each sample confidence score of the particle filter, while the SGM and PGM are used to solve the data association problem. Each tracker also contains a sample selection scheme to update the appearance model with high confidence key samples. Finally, the data association module is used to construct the similarity matrix S to match detections, $d_t \in \mathcal{D}^t$, with existing trackers, $b_t \in \mathcal{B}_e^t$, at time t. Furthermore, it determines initialization, termination and on-hold states of the trackers, and supports the tracker with key samples from the target trajectory.

In this chapter, we used the fast pedestrian detector (FPD) [32] for multi-person tracking. In Section 4.3, we used other pre-trained detectors, such as the vehicle detector proposed in Chapter 3 [81, 82], and the method in [33] to measure the tracking performance on several detection conditions and different types of objects.

4.2 Tracking Scheme

Each object tracker is based on the particle filter tracking framework that uses the sparse representations and 2DPCA as the appearance model. We incorporate two

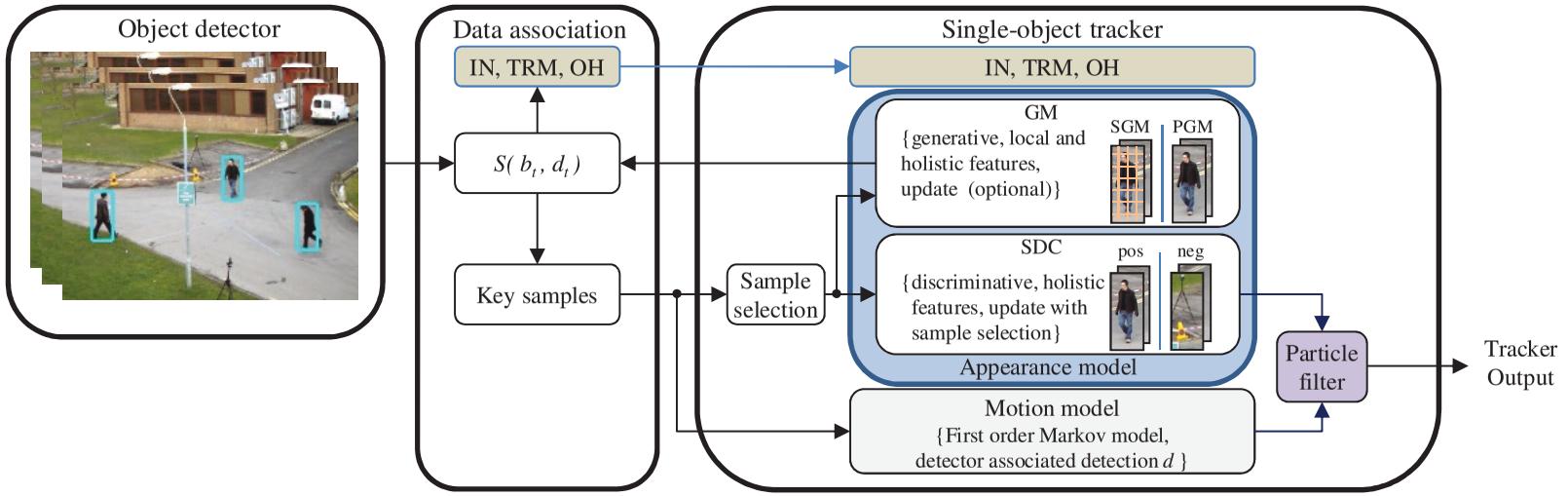


Figure 4.1: Block diagram of the proposed multi-object tracking scheme, where IN, TRM, OH, pos, and neg denote initialization, termination, on-hold, positive, and negative, respectively (see text for details).

measurements from the detector and tracker into the particle filter, and propose a novel collaborative model that directly affects the likelihood function to obtain the posterior estimate of the target location. We construct the appearance model of the target by using discriminative and generative appearance models, for the likelihood function and the data association. In the following, we use a gate function $\mathcal{G}_{b_t}$ to represent the state of the tracker b_t when associated to the detection d_t at time t. The gate function is defined as

$$\mathcal{G}_{b_t} = \begin{cases} 1, & \text{if } b_t \text{ is associated with } d_t \text{ at time } t \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

4.2.1 Particle Filter using the Robust Collaborative Model

In the Bayesian tracking framework, the posterior at time t is approximated by a weighted sample set $\{\mathbf{x}_t^i, \boldsymbol{w}_t^i\}_{i=1}^{N_s}$, where $\boldsymbol{w}_t^i$ is the weight of particle, $\mathbf{x}_t^i$, and N_s is the total number of particles. The state $\mathbf{x}$ consists of translation (x, y), average velocity (v_x, v_y), scale $\hat{s}$, rotation angle θ, aspect ratio η, and skew direction ϕ.

In the proposed method of tracking, we adopt a *hybrid motion model* based on the first-order Markov chain and the associated detection. The new candidate state $\mathbf{x}_t^{d_t}$ at time t is provided to the motion model if a detection is successfully associated to the tracker (i.e., $\mathcal{G}_{b_t} = 1$), and the initial velocity is set to be the average velocity of the tracker particles. The candidate state at time t, $\mathbf{x}_t$, relates to the set of propagated particles X^{b_t} and the set of associated detection X^{b_t,d_t} by

$$\mathbf{x}_t = \begin{cases} \mathbf{F}\mathbf{x}_{t-1} + \mathbf{x}_Q & \text{if } \mathbf{x}_t \in X^{b_t} \\ \mathbf{x}_t^{d_t} + \mathbf{x}_P & \text{if } \mathbf{x}_t \in X^{b_t,d_t} \end{cases} \tag{4.2}$$

where $\mathbf{x}_Q$ and $\mathbf{x}_P$ are the Gaussian noise vectors, $N_s = N_s^P + N_s^\Gamma$, and N_s^P and N_s^Γ are the cardinality of X^{b_t} and X^{b_t,d_t}, respectively. In the above equation, $\mathbf{F}$ denotes

the transition matrix of size 8×8, which is defined as

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

The *measurement model* of the proposed particle filter consists of two types. The first type is available every time t from the propagated particles $\mathbf{z}_{1:t}^{b_t}$. The second type is from the newly created particles $\mathbf{z}_t^{d_t}$ that are available at time t when a detection window, d_t, is associated to a tracker, b_t (i.e., $\mathcal{G}_{b_t} = 1$). Since it is difficult to sample particles from the posterior distribution directly, we use an importance density [108, 109] to obtain the candidate samples, $\mathbf{x}_t^i$, from this distribution. In the proposed scheme, when the tracker b_t is associated to a detection d_t at a given time t, then the candidate particles are sampled from the importance distribution, $q(\mathbf{x}_t^i|\mathbf{x}_{1:t-1}^i, \mathbf{z}_t^{b_t}, \mathbf{z}_t^{d_t})$, that depends on the previous states, $\mathbf{x}_{1:t-1}^i$, and the two types of measurements, $\mathbf{z}_t^{b_t}$ and $\mathbf{z}_t^{d_t}$. The posterior probability of the candidate location, given the available measurements, can be approximately expressed as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}^{b_t}, \mathbf{z}_t^{d_t}) \approx \sum_{i=1}^{N_s} \boldsymbol{w}_t^i \delta(\mathbf{x}_t - \mathbf{x}_t^i) \tag{4.4}$$

where

$$\boldsymbol{w}_t^i \propto \boldsymbol{w}_{t-1}^i \frac{p(\mathbf{z}_t^{b_t}, \mathbf{z}_t^{d_t}|\mathbf{x}_t^i) p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{x}_t^{d_t})}{q(\mathbf{x}_t^i|\mathbf{x}_{1:t-1}^i, \mathbf{z}_t^{b_t}, \mathbf{z}_t^{d_t})} \tag{4.5}$$

and $p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{x}_t^{d_t})$ is the transition probability. In the proposed method, the particles

are resampled every time t, and then we have $\boldsymbol{w}_{t-1}^i = 1/N_s, \forall i$, and we ignore $\boldsymbol{w}_{t-1}^i$ term. Let the importance density be proportional to the prior as

$$q(\mathbf{x}_t^i|\mathbf{x}_{1:t-1}^i, \mathbf{z}_t^{b_t}, \mathbf{z}_t^{d_t}) \propto p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{x}_t^{d_t}) \tag{4.6}$$

Using (4.6), (4.5) reduces to

$$\boldsymbol{w}_t^i \propto p(\mathbf{z}_t^{b_t}, \mathbf{z}_t^{d_t}|\mathbf{x}_t^i) \tag{4.7}$$

In the current frame, since the propagated particles sampled at time t corresponding to the tracker position in the previous frame and the particles sampled at time t from the associated detection are independent, the particle weights satisfy

$$\boldsymbol{w}_t^i \propto \begin{cases} p(\mathbf{z}_t^{b_t}|\mathbf{x}_t^i) & \text{if } \mathbf{x}_t^i \in X^{b_t} \\ p(\mathbf{z}_t^{d_t}|\mathbf{x}_t^i) & \text{if } \mathbf{x}_t^i \in X^{b_t,d_t} \end{cases} \tag{4.8}$$

where $p(\mathbf{z}_t^{b_t}|\mathbf{x}_t^i)$ and $p(\mathbf{z}_t^{d_t}|\mathbf{x}_t^i)$ are the likelihoods of the i^{th} candidate state $\mathbf{x}_t^i$, in case of $\mathbf{x}_t^i$ belongs to the set of propagated particles X^{b_t} and that of the set of newly created ones X^{b_t,d_t}, respectively. By normalizing the particle weights, the resulting state estimate is represented as a weighted average of the candidate locations. This makes the proposed scheme more robust to noisy detection results compared to maximum a posteriori methods.

When there is no detection associated with a tracker (i.e., $\mathcal{G}_{b_t} = 0$), the proposed particle filter reduces to the bootstrap particle filter [74, 109]. In such a case, the particle weights satisfy [109]

$$\boldsymbol{w}_t^i \propto p(\mathbf{z}_t^{b_t}|\mathbf{x}_t^i) \tag{4.9}$$

a) Robust Collaborative Model

The object detector applies computationally expensive space-scale search to the entire image to localize specific class of objects, and proposes candidate locations that have high probability of existence. To exploit high confidence associated detections, we incorporate a set of new particles, X^{b_t,d_t}, in the likelihood function, to allow the object detector to guide the trackers. Let $H_{SDC}(\mathbf{x}_t^i)$ denote SDC tracker confidence score of candidate $\mathbf{x}_t^i$. The likelihood of the measurement, $\mathbf{z}_t$, can be computed by

$$p(\mathbf{z}_t|\mathbf{x}_t^i) = \pi^i H_{SDC}(\mathbf{x}_t^i) \tag{4.10}$$

where

$$\pi^i = \begin{cases} 1-\gamma_{CF} & \text{if } \mathcal{G}_{b_t} = 1, \mathbf{x}_t^i \in X^{b_t} \\ \gamma_{CF} & \text{if } \mathcal{G}_{b_t} = 1, \mathbf{x}_t^i \in X^{b_t,d_t} \\ 1 & \text{otherwise, i.e., } \mathcal{G}_{b_t} = 0 \end{cases} \tag{4.11}$$

and $\gamma_{CF} \in [0,1]$ is the collaborative factor. In (4.10), the particles from the associated detections and previously propagated particles are weighted differently. Figure 4.2 shows the effect of changing the collaborative factor value. Figure 4.3(a) and (b) show an example of particle weights for the detector particles and the propagated particles using $\gamma_{CF} = 0.54$. If $\mathcal{G}_{b_t} = 1$ and $\gamma_{CF} > 0.5$, the weight π^i allows the detector to guide the tracker by giving more weights to the newly associated particles than the propagated particles. However, a detector may have false positives, and thus, the tracker should not depend completely on the detector. From our experiments, we find that the proposed scheme with the value of γ_{CF} between 0.5 and 0.85 performs best. If the detector suffers from missing detections (i.e., $\mathcal{G}_{b_t} = 0$), the likelihood function in (4.10) will only depend on the previously propagated particles $\mathbf{x}_t^i \in X^{b_t}$, which represent the bootstrap particle filter [74]. Our collaborative model is based on the hybrid

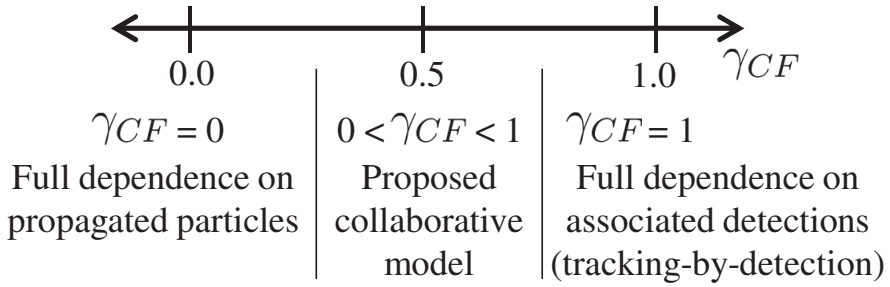


Figure 4.2: Effect of changing the collaborative factor γ_{CF}.

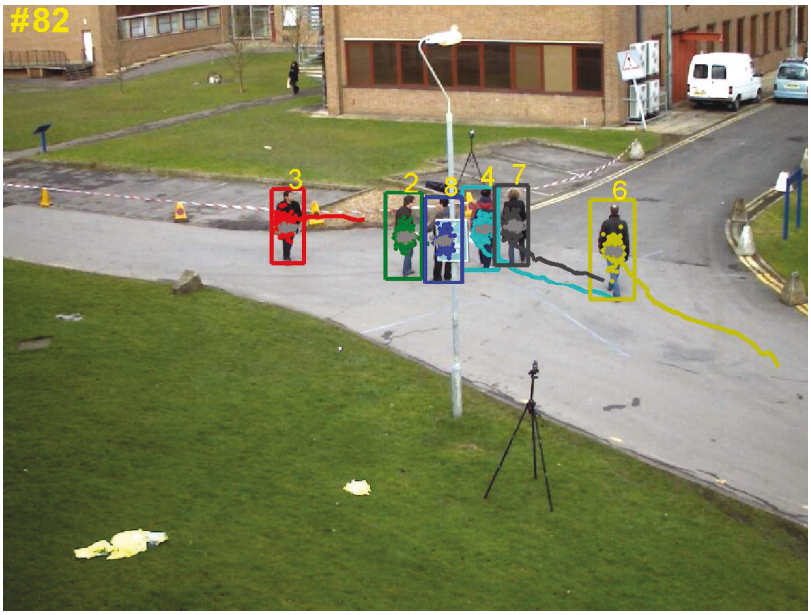
motion model that incorporates associated detections with object dynamics. In contrast, the motion model adopted in [5] depends only on propagated particles, and the likelihood function depends on tracker appearance model and the detector confidence density. The collaborative model in [61] only exists in the proposal distribution and the likelihood is without weighting collaborative factor.

b) Resampling

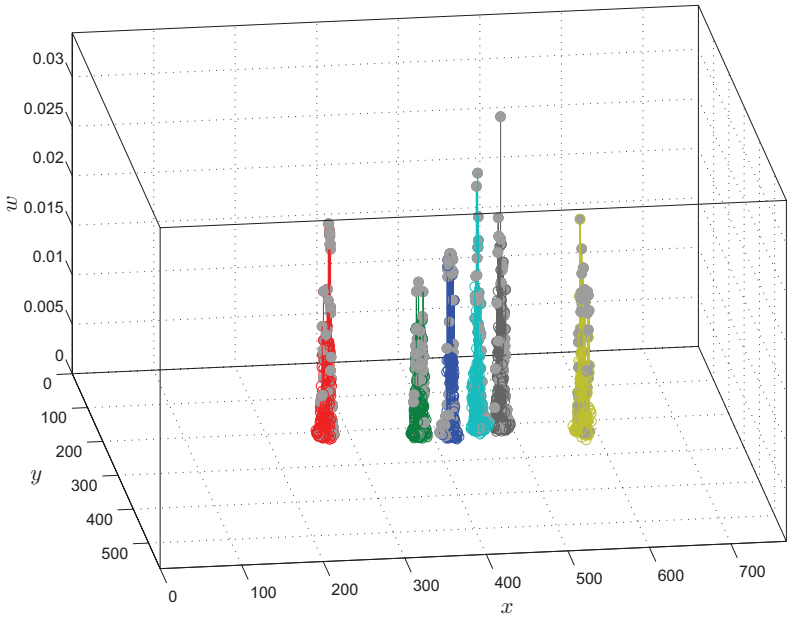
In each frame, the set of candidate particles $\{\mathbf{x}_t^i, \boldsymbol{w}_t^i\}_{i=1}^{N_s}$ are resampled to avoid the degeneracy problem. The resampling process also allows the detector to guide the tracker effectively. As each tracker resamples particles based on particle weights computed from the proposed collaborative model (4.10), the propagated particles with low weights are replaced with newly created particles from the associated detections.

4.2.2 Appearance Model

In the proposed method, the SGM and SDC are used in a way different from that in [110]. First, we do not use the collaboration between SGM and SDC [110], instead we use SGM with PGM to compute the similarity matrix of the data association module for occlusion handling (4.23), and the modified SDC model is used to compute the likelihood of the particle filter (4.10). The number of particles in the filter



(a)



(b)

Figure 4.3: Effect of the proposed collaborative model on the tracker particles. (a) Illustrates the candidate particles proposed by the object detector (masked as gray) and propagated particles (colored). (b) Particles weights for new (masked as gray) and propagated particles (colored).

is usually larger than the number of detections and trackers at time t, and the computational complexity of SDC is lower than SGM. Therefore, the resulting tracker is more efficient. Second, our SDC uses the downsampled grayscale image without the feature selection method used in [110]. Third, our SDC confidence measure depends on the sparsity concentration index [111]. Finally, we propose the key sample selection scheme to update the appearance models with high confidence samples.

a) Sparsity-based Discriminative Classifier

We construct a discriminative sparse appearance model to compute the confidence score as used in (4.10). The initial training samples are collected in a similar way to [110], where each SDC tracker is initialized using N_p positive samples drawn from the object center with a small variation from the center of the detection state $\mathbf{x}_t^d$, and N_n negative samples are taken from the annular region surrounding the target center without overlap with a detection window d_t. Next, each sample is normalized to a canonical size of $(m_1 \times m_2)$, and vectorized to be one column of the matrix $\mathbf{A} \in \mathbb{R}^{r \times N^t}$, where $r = mn$ and $N^t = N_p + N_n + N_{p,u}^t + N_{n,u}^t$, such that $N_{p,u}^t$ and $N_{n,u}^t$ denote the buffer size of the selected key samples up to time t. Let the measurement corresponding to the candidate location $\mathbf{x}_t^i$ be denoted by $\mathbf{z}_t^i \in \mathbb{R}^r$. We obtain the sparse coefficients $\tilde{\alpha}^i$ for the i^{th} candidate by solving the following optimization problem,

$$\min_{\tilde{\alpha}^i} \left\| \mathbf{z}_t^i - \mathbf{A}\tilde{\alpha}^i \right\|_2^2 + \lambda_{SDC} \left\| \tilde{\alpha}^i \right\|_1 \tag{4.12}$$

We compute the classifier confidence score by

$$H_{SDC}(\mathbf{x}_t^i) = \exp\left(-\frac{(\varepsilon_+^i - \varepsilon_-^i)}{\sigma} \right) \Omega_{SCI}(\tilde{\alpha}^i) \tag{4.13}$$

where $\varepsilon_+^i = \left\| \mathbf{z}_t^i - \mathbf{A}_+ \tilde{\alpha}_+^i \right\|_2^2$ is the reconstruction error of the candidate $\mathbf{z}_t^i$ with respect to the template set of the positive class $\mathbf{A}_+$, and the sparse coefficient vector of the i^{th} candidate that corresponds to the positive class, $\tilde{\alpha}_+^i$. Similarly, $\varepsilon_-^i = \left\| \mathbf{z}_t^i - \mathbf{A}_- \tilde{\alpha}_-^i \right\|_2^2$

is the reconstruction error of the same candidate $\mathbf{z}_t^i$ with respect to the template set of the negative class $\mathbf{A}_-$, and the corresponding sparse coefficient vector $\tilde{\alpha}_-^i$. The parameter σ adjusts the confidence measure, and $\Omega_{SCI}(\tilde{\alpha}^i)$ represents the sparsity concentration index (SCI) [111] defined as

$$\Omega_{SCI}(\tilde{\alpha}^i) = \frac{J \cdot \max_j \|\delta_j'(\tilde{\alpha}^i)\|_1 / \|\tilde{\alpha}^i\|_1 - 1}{J - 1} \in [0, 1] \tag{4.14}$$

where δ_j' is a function that selects the coefficients corresponding to the j^{th} class and suppresses the rest, and J is the number of classes ($J = 2$ in this work). The SCI checks the validity of a candidate such that it can be represented by a linear combination of the training samples in one class. When the sparse coefficients concentrate in a certain class, the SCI value is high. This index allows each tracker to assign high weights to candidates resembling the positive training samples, and rejects others related to other targets or background structures.

The SDC tracker is updated every R_u frames using the selected key samples, K_u^t (Section 4.2.3). At each key sample location, we collect positive and negative samples as part of the initialization process. To leverage between computational load and memory requirement, we set the maximum number of positive and negative samples. If the number of positive, $N_{p,u}^t$ or negative, $N_{n,u}^t$ samples exceeds the limit, we replace the old samples (other than those collected in the first frame) with the new selected key samples.

b) Sparsity-based Generative Model

We use a sparsity-based generative model to measure similarity in the data association module. Figure 4.4 illustrates the block diagram of the proposed SGM in the training and test modes. The training template consists of M local patches, $\{\mathbf{y}_i\}_{i=1}^M$ and

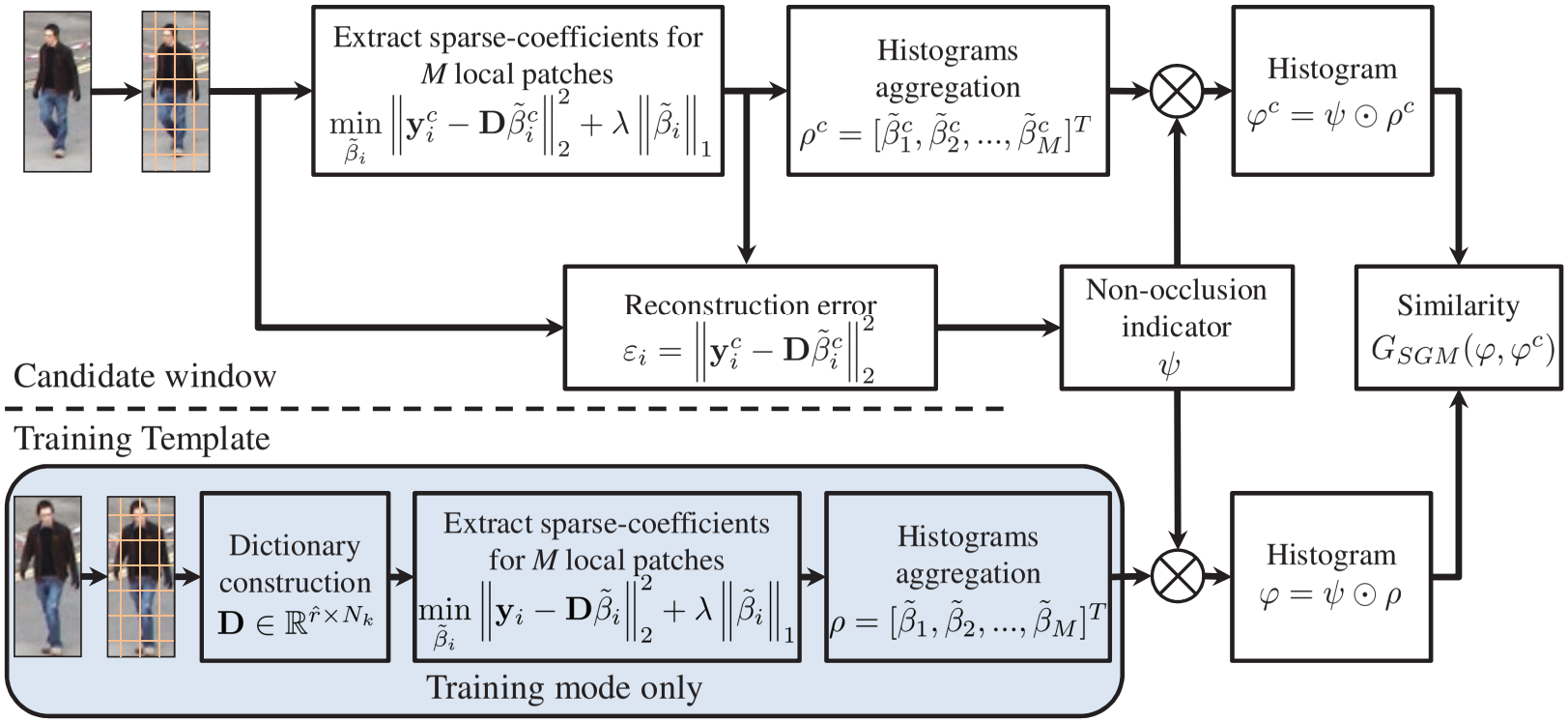


Figure 4.4: Block diagram of the sparsity-based generative model.

each patch of size $\hat{m}_1 \times \hat{m}_2$. These M patches are vectorized[1] and quantized into N_k centroids using the k-means algorithm to construct the dictionary $\mathbf{D} \in \mathbb{R}^{\hat{r} \times N_k}$ ($\hat{r} = \hat{m}_1 \hat{m}_2$). For the i^{th} patch, $\mathbf{y}_i$, the sparse-coefficients, $\tilde{\beta}_i \in \mathbb{R}^{N_k \times 1}$, is computed by

$$\min_{\tilde{\beta}_i} \left\| \mathbf{y}_i - \mathbf{D}\tilde{\beta}_i \right\|_2^2 + \lambda_{SGM} \left\| \tilde{\beta}_i \right\|_1 \tag{4.15}$$

The adopted SGM is concerned with representing the appearance of the positive class of the tracker by using the sparse coefficients of M local patches of the object and candidate location c, where each location is represented by a sparse histogram feature vector $\rho = [\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_M]^T$, and $\rho^c = [\tilde{\beta}_1^c, \tilde{\beta}_2^c, \ldots, \tilde{\beta}_M^c]^T$, corresponding to the initial object and the candidate location, respectively. To handle occlusion, the patch reconstruction error, $\{\varepsilon_i = \left\| \mathbf{y}_i^c - \mathbf{D}\tilde{\beta}_i^c \right\|_2^2\}_{i=1}^M$, is used to suppress the coefficients of occluded patches. Let ψ_i be the non-occlusion indicator for the i^{th} patch and is computed by

$$\psi_i = \begin{cases} \mathbf{1}_{N_k,1} & \text{if } \varepsilon_i < \varepsilon_0 \\ \mathbf{0}_{N_k,1} & \text{otherwise} \end{cases} \tag{4.16}$$

where $\mathbf{1}_{N_k,1}$, and $\mathbf{0}_{N_k,1}$ denote the vector of size N_k of ones and zeros. The final histogram can be represented by $\varphi = \psi \odot \rho$, and $\varphi^c = \psi \odot \rho^c$, corresponding to the training template, and the candidate location, where $\odot$ denotes the element-wise multiplication. By taking the spatial representation into consideration, the resulting histogram, φ can handle occlusion effectively. Figure 4.5 illustrates the effect of the partial occlusion handling scheme. If the reconstruction error is greater than the threshold, ε_0, then the non-occlusion indicator, ψ, suppresses these patches. The generative model similarity, $G_{SGM}(b_t, c)$, between the candidate φ_c and the model φ is measured by using the intersection kernel.

As in [110], the dictionary, $\mathbf{D}$, is fixed during the tracking process, while the

[1]The vectorization function is defined as Mat2Vec: $\mathbb{R}^{m_1 \times m_2} \rightarrow \mathbb{R}^r$, where $r = mn$ is the dimension of the vector, and $(m_1 \times m_2)$ is the order of the input matrix. The inverse of the vectorization function is defined as Vec2Mat: $\mathbb{R}^r \rightarrow \mathbb{R}^{m_1 \times m_2}$.

(a) (b) (c)

Figure 4.5: Sample results for SGM partial occlusion handling scheme, where the marked patches with the same tracker color are the patches at which SGM reconstruction error is greater than the SGM error threshold.

sparse histogram of the initial template, $\rho_{initial}$, is updated every update rate, R_u. The sparse histogram is updated by

$$\rho_{new} = \mu\rho_{initial} + (1-\mu)\rho_K \tag{4.17}$$

where μ is the learning rate, and ρ_K represents the sparse histogram corresponding to the selected key sample from the set K_u^t that provides the maximum similarity to the training templates (see Section 4.2.3 for the sample selection scheme). This conservative update scheme by using the confidence key samples and maintaining the initial template provide effective tracking.

c) 2DPCA-based Generative Model

In addition to part-based SGM, we use a holistic generative model based on the 2DPCA scheme [94], referred to as PGM, to solve the data association problem. The reason being that a combination of PGM and SGM increases the tracking performance (see Section 4.3). For each tracker b_t, we use N positive samples, $\{\mathbf{Y}_j\}_{j=1}^N$ each of size $m_1 \times m_2$, where samples are taken from the positive class of the initial target location, or selected key samples, K_u^t.

The image covariance matrix **Cov** is defined by

$$\mathbf{Cov} = \frac{1}{N}\sum_{j=1}^{N}(\mathbf{Y}_j - \bar{\mathbf{Y}})^\top(\mathbf{Y}_j - \bar{\mathbf{Y}}) \tag{4.18}$$

where $\bar{\mathbf{Y}}$ is the average image of all training samples, and **Cov** is the nonnegative definite matrix. The objective of 2DPCA is to find the optimal orthonormal matrix, $\mathbf{V}_{opt}$, that maximizes the total scatter in the learned subspace. The total scatter criterion $\mathbf{J}(\mathbf{V})$ is defined by

$$\mathbf{J}(\mathbf{V}) = \mathbf{V}^\top\mathbf{CovV} \tag{4.19}$$

The optimal projection matrix $\mathbf{V}_{opt}$ is composed of the r_1 eigenvectors of matrix **Cov** corresponding to the first r_1 largest eigenvalues, where the vectors are stacked together in matrix $\mathbf{V}$ of size $m_2 \times r_1$. We extract features of the j^{th} training example, $\mathbf{Y}_j$, through projecting on matrix $\mathbf{V}$, as $\mathbf{F}^j = \mathbf{Y}_j\mathbf{V}$, and then we vectorize the resulting feature matrix and have the feature vector $\mathbf{f}^j$ of size $1 \times m_1 r_1$.

For each candidate location, we project the candidate sample, $\mathbf{Y}^c$, using the matrix $\mathbf{V}$, and vectorize the resulting matrix to obtain the test feature vector $\mathbf{f}^c$ of size $1 \times m_1 r_1$. The nearest neighbor classifier is used to infer the index of the j^{th} training example, $\hat{j}$ closest to the test vector $\mathbf{f}^c$

$$\hat{j} \leftarrow \underset{j\in\{1,2,\ldots,N\}}{\operatorname{argmin}} \|\mathbf{f}^c - \mathbf{f}^j\|_2 \tag{4.20}$$

where $\|.\|_2$ denotes the l_2-norm. The reconstruction error between the test image and the training examples is $\varepsilon_{PGM} = \|\mathbf{a}_{\hat{j}} - \mathbf{a}_c\|_2$, where $\mathbf{a}_{\hat{j}} = \text{Mat2Vec}(\mathbf{F}^{\hat{j}}\mathbf{V}^\top)$ and $\mathbf{a}_c = \text{Mat2Vec}(\mathbf{F}^c\mathbf{V}^\top)$. The similarity between the test and training features is computed by

$$G_{PGM} = \exp(-\varepsilon_{PGM}/\hat{\sigma}^2) \tag{4.21}$$

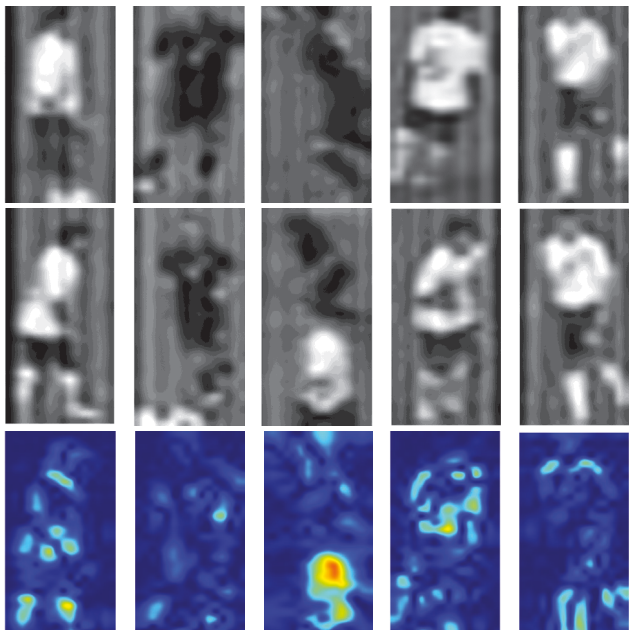


Figure 4.6: (Top) Reconstructed nearest neighbor training samples by PGM. (Middle) Reconstructed patches at candidate locations. (Bottom) Absolute reconstruction error, where the pixel with brighter color means high error value.

Figure 4.6 shows a sample intermediate output from the proposed PGM scheme. The PGM is able to retrieve the closest training patches in 2DPCA feature subspace, which provides accurate similarity measures in (4.21).

Similar to SDC tracker, PGM is updated every R_u frames, by using the initial positive and the selected key samples at time t, where $N = N_p + N_{p,u}^t$. To update 2DPCA feature space, we used a batch learning technique. In this scheme, we update the optimal projection matrix, $\mathbf{V}_{opt}$, and extract the feature vectors, $\{\mathbf{f}^j\}_{j=1}^N$. While the incremental 2DPCA learning has been used in [112], we find that batch learning performs more efficiently than the incremental learning scheme, since we replace some samples every update rate with newly selected key samples.

4.2.3 Sample Selection

We propose a sample selection scheme to learn and adapt the appearance model for each tracker by using the samples with high confidence from the object trajectory, in a way similar to existing methods [5, 63, 113]. Examples for key sample locations in the object trajectory are shown in Figure 4.7, where two scenarios for the key samples are selected from the tracker history. The sample selection scheme alleviates the problem of including occluded samples for more effective model update and thus, reduces the drifting problem. The proposed sample selection scheme is based on the following criteria:

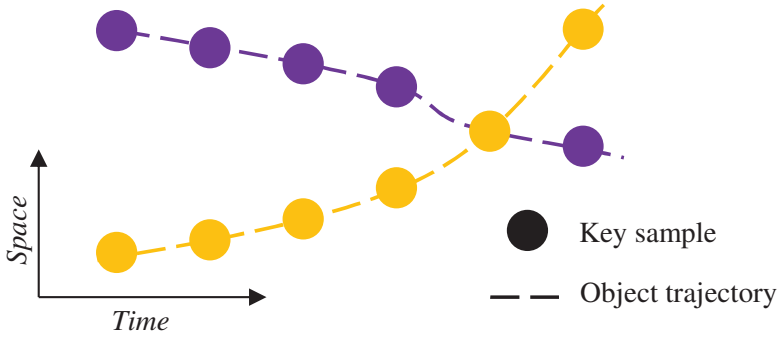
1. We measure the goodness of the key samples. A good key sample is one at which the tracker b_t does not intersect with other trackers or nearby detections except the associated detection d_t. We denote the set of good key samples at time t by K_g^t.

2. We use the online trained SDC tracker to measure the similarity between the current appearance model of the tracker, b_t, and the i^{th} good key sample $K_{g,i}^t \in K_g^t$ by

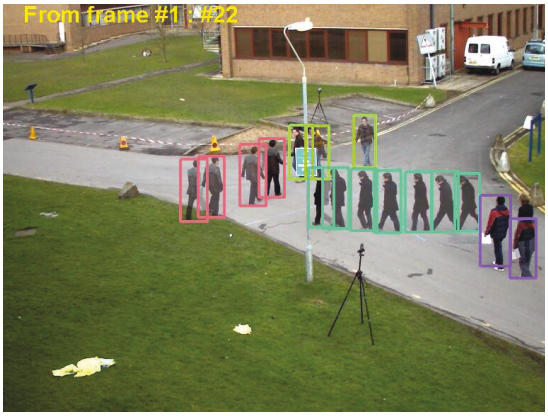
$$S_{DC}(b_t, K_{g,i}^t) = \exp(-(\varepsilon_+^i - \varepsilon_-^i)/\sigma^2) \tag{4.22}$$

   where $\varepsilon_+^i = \left\|\mathbf{z}_t^i - \mathbf{A}_+\tilde{\alpha}_+^i\right\|_2^2$, $\varepsilon_-^i = \left\|\mathbf{z}_t^i - \mathbf{A}_-\tilde{\alpha}_-^i\right\|_2^2$, and $\tilde{\alpha}_+^i$ and $\tilde{\alpha}_-^i$ are computed by using (4.12).

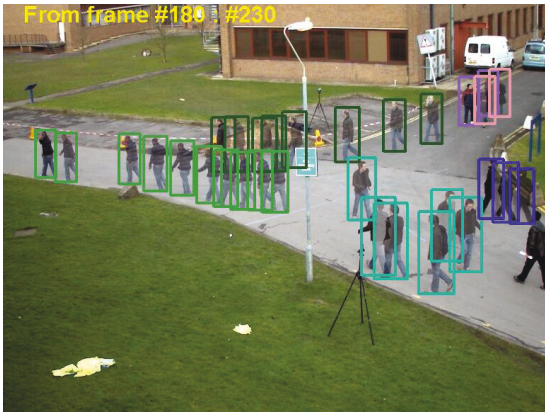
3. If $S_{DC}(b_t, K_{g,i}^t) > s_0 \geq 0$, where s_0 is the SDC similarity threshold, then this key sample is selected for the model update. The final set of selected key samples, K_u^t, which have high similarity with the SDC tracker, are used to update the tracker appearance model (Section 4.2.2). It should be observed that when $s_0 = 0$, all the samples are selected.



(a)



(b)



(c)

Figure 4.7: (a) Key samples in the object trajectories and occlusion issues that should be handled, (b and c) Examples for key samples selected from object trajectories, using a sequence from the *PETS09-S2L1* dataset.

4.2.4 Data Association

The similarity matrix S for data association measures the relation between a tracker $b_t \in \mathcal{B}_e^t$ and a detection $d_t \in \mathcal{D}^t$ by

$$S(b_t, d_t) = G(b_t, d_t)O(b_t, d_t) \tag{4.23}$$

where $G(b_t, d_t) = G_{SGM}(b_t, d_t) + G_{PGM}(b_t, d_t)$ considers the appearance similarity between the tracker b_t and detection d_t, and $O(b_t, d_t)$ represents the overlap ratio between the tracker and the detection to suppress confusing detections, where the overlap ratio is based on the PASCAL VOC criterion [105].

The association is computed online by using the Hungarian algorithm [114] to match a tracker to a detection in a way similar to existing methods [5, 63]. The proposed data association scheme iteratively finds the maximum in the matrix S, and associates the tracker b_t to a detection d_t if $S(b_t, d_t)$ is larger than a threshold s_1. The row and the column corresponding to $S(b_t, d_t)$ are removed. As the object detector is likely to miss some objects, using the similarity threshold, s_1, can alleviate the tracker to be updated with confusing nearby detections. Furthermore, we select a number of key samples to update the appearance model (Sections 4.2.3 and 4.2.2). We initialize new trackers with non-associated detection windows if the maximum overlap with other existing trackers is less than O_1 to avoid creating multiple trackers for the same target.

Re-detection Module

A pre-trained object detector usually suffers from false positives and negatives, thereby causing trackers to drift. On the other hand, a tracker does not perform well in the presence of heavy occlusion or background clutters. To handle these challenging cases, we introduce the inactive or on-hold states before tracker termination in case the tracker misses a high number of detections.

Let the set of trackers on-hold be denoted as $\mathcal{B}_h^t$. When the tracker does not estimate the target location at an inactive state, we adopt the PGM (Section 4.2.2) to measure the similarity between the tracker on-hold $b_t \in \mathcal{B}_h^t$ and the new candidate location. When the tracker is in the inactive state b_t^h, it still can be reinitialized after checking the similarity with the new un-associated detection, d_t^u by $S_h(b_t^h, d_t^u) = G_{PGM}(b_t^h, d_t^u)$ (where G_{PGM} is computed by (4.21)). The inactive tracker is reactivated if $S_h(b_t^h, d_t^u) > s_2$, where s_2 is a pre-defined threshold. During the inactive state, the proposed tracker can re-identify lost targets and discriminate among trackers using the 2DPCA feature space learned from selected key samples.

4.3 Experimental Results

4.3.1 Datasets

We evaluate the tracking performance of the proposed algorithm using seven challenging sequences, namely, the *PETS09-S2L1, PETS09-S2L2* [87], *UCF Parking Lot (UCF-PL)* dataset [63], *Soccer* dataset [62], *Town Center* dataset [88], and *Urban* as well as *Sunny* sequences from *LISA 2010* dataset [84], and compare it with that of several state-of-the-art online multi-object tracking methods.

The *PETS09-S2L1* sequence consists of 799 frames of 768×576 pixels recorded at 7 frames per second with medium crowd density. The *PETS09-S2L2* sequence consists of 442 frames with the same resolution and frame rate as the *PETS09-S2L1* sequence, but it contains heavy crowd density and illumination changes. The target objects undergo scale changes, long-term occlusion, and with similar appearance. The ground truth (GT) data from [115, 116] and [117] are used for evaluating the tracking results on *PETS09-S2L1* and *PETS09-S2L2*, respectively. The *Soccer* sequence consists of 155 frames of 960×544 pixels recorded at 3 to 5 frames per second. The challenging factors of this sequence include heavy occlusion, sudden change of motion direction of players, high similarity among players of the same team, and scale changes. The GT data provided by [62] are used for evaluation. On the *PETS09-S2L1*, *PETS09-S2L2*

and *Soccer* sequences, the FPD detector [32] is used as the baseline detector for the proposed tracking scheme.

The *UCF-PL* dataset consists of 998 frames of 1920×1080 pixels recorded at 29 frames per second with medium crowd density, long-term occlusion, and targets of similar appearance. On this dataset, the detection results of the part-based pedestrian detector proposed in [63] are used for evaluation based on the GT data provided by [118].

The *Town Center* dataset consists of 4500 frames of 1080×1920 pixels recorded at 25 frames per second. The dataset contains medium crowd density, heavy occlusion, and scale changes. In [88], two categories of GT annotations are provided based on the full body and head regions of pedestrians. On this dataset, the aggregated channel feature (ACF) detector proposed by Dollár *et al.* [33] is used for performance evaluation. In the case of the full body of pedestrians, it has been observed that the ACF detector does not perform well on this sequence as the false positive rate is high. To alleviate this problem, the first 500 frames of this sequence are used to collect hard-negative samples related to the background clutters, and the ACF detector is re-trained using both the *INRIA* dataset [6] and hard-negative samples. In case of tracking multiple people based on the head regions, the positive training examples provided in [88] and negative samples collected from the first 500 frames of this sequence are used to train the ACF detector.

The *Urban* and *Sunny* sequences from the *LISA 2010* dataset [84] contain car images of 704×480 collected at 30 frames per second from a camera mounted on a moving vehicle. The *Urban* sequence (300 frames) was captured from an urban area with a low traffic density on a cloudy day, while the *Sunny* sequence (300 frames) was captured from a highway with medium traffic density on a sunny day. The challenging factors of these sequences include the effect of camera vibration, illumination changes, and the targets' scale changes; the GT data are provided by [84]. The pre-trained vehicle detector proposed in [81, 82] is used for evaluation on this dataset.

4.3.2 Qualitative Results

In this section, we study the qualitative performance of the proposed tracking scheme using the datasets mentioned above. Figures 4.8 and 4.9 show some of the tracking results and videos are available at `https://users.encs.concordia.ca/~rcmss/`.

PETS09-S2L1: Figure 4.8(a) shows the sample tracking results of the proposed scheme on the *PETS09-S2L1* sequence. The proposed method performs well despite several short-term occlusions, scale and pose changes. Furthermore, it should be mentioned that the pre-trained FPD detector [32] misses objects that are close to the camera or those located far from the camera.

PETS09-S2L2: Figure 4.8(b) shows that non-occluded targets are tracked well although targets with long-term occlusions or located far from the camera are missed. Again, it should be mentioned that the FPD detector [32] misses numerous detections in this sequence due to the high crowd density.

Soccer: This sequence contains soccer players with similar visual appearance and fast motion. The FPD detector [32] is not trained to detect the soccer players at different poses. Nevertheless, the proposed scheme performs well with accurate short tracklets, as shown in Figure 4.8(c).

UCF-PL: This sequence contains crowds of medium density, with occlusions. Figure 4.8(d) shows some tracking results for the proposed scheme using the detector in [63]. Despite the challenges of the sequence, the proposed tracking scheme maintains long trajectories.

Town Center: The crowd density of this sequence is medium with a number of long-term occlusions. Figures 4.8(e) and (f) show sample tracking results corresponding to full body and head, respectively. While it is difficult to track the full human body due to heavy occlusions, or the head due to false positives, the proposed method performs well.

Figure 4.8: Sample tracking results for five sequences, the arrangement from top to bottom as (a) and (b) *PETS09-S2L1*, and *PETS09-S2L2*, respectively, (c) *Soccer* sequence, (d) *UCF-PL* sequence, (e) *Town Center* dataset (body), and (f) *Town Center* dataset (head).

LISA 2010: Figures 4.9(a) and (b) show the sample results of our tracker using the detector in [81, 82] on the *Urban* and *Sunny* sequences. The *Urban* sequence contains only one vehicle, but there is illumination change and the effect of camera vibrations. The *Sunny* sequence contains, on average, three non-occluded vehicles with different velocities. In spite of these challenges, the proposed scheme tracks the vehicles very well in both cases.

4.3.3 Quantitative Results

We use the CLEAR MOT metrics [119] including multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), false negative rate (FNR), false positive rate (FPR), and identity switches (IDSW) for evaluating the performance of the proposed tracker. We use the overlap threshold of 0.5 for all experiments. For this study, we set the various parameters to be $N_s^P = 150$, $N_s^\Gamma = 100$, $N_p = N_{p,u}^t = 10$, $N_n = N_{n,u}^t = 20$, $R_u = 10$, $\lambda_{SDC} = 0.02$, $\lambda_{SGM} = 0.01$, $\hat{\sigma} = 10^4$, $\varepsilon_0 = 0.8$, $\mu = 0.6$, $\hat{\sigma} = 5 \times 10^6$, $s_0 = 1.0$, $s_1 = 2.5$, $s_2 = 0.7$, and $O_1 = 0.2$. For the multi-person tracking sequences, namely, *PETS09-S2L1, PETS09-S2L2*, *UCF-PL*, *Soccer*, and *Town Center (Body)*, we use $m_1 = 32$, $m_2 = 16$, $M = 84$, $\hat{m}_1 = \hat{m}_2 = 6$ and $N_k = 50$. Further, for the multi-head tracking sequence, namely, *Town Center (Head)*, as well as the multi-vehicle tracking sequences, namely, *Urban* and *Sunny*, we use $m_1 = m_2 = 16$, $M = 16$, $\hat{m}_1 = \hat{m}_2 = 6$ and $N_k = 16$.

Effect of the Collaborative Factor: To measure the effect of the proposed collaborative model, we changed the value of the collaborative factor γ_{CF} in the interval $[0, 1]$ in increments of 0.2. Figure 4.10 shows the performance of the proposed method with different values of γ_{CF} for the *PETS09-S2L1* sequence. When $\gamma_{CF} = 0$, the likelihood function of the particle filter is based completely on the propagated particles, and the proposed method does not perform well due to the degeneracy problem. When $\gamma_{CF} = 1$, the likelihood function is based on the associated detections, and the

Figure 4.9: Sample tracking results for *LISA 2010* dataset, where (a) and (b) correspond to *Urban* and *Sunny* sequences, respectively.

tracker does not perform well due to false positives and missed detections. The proposed method performs best for this sequence when $\gamma_{CF} = 0.8$, as can be seen from Figure 4.10. It is worth noting that for high tracking performance, the value of γ_{CF} should be adjusted according to the detector used. For detectors with high precision and recall (the ones used in the *PETS09-S2L1*, *UCF-PL*, *Town Center (Head)*, *Urban* and *Sunny* sequences), the proposed tracker provides a high MOTA value when γ_{CF} is in the interval of $[0.65, 0.85]$. On the other hand, when the detector has low precision and recall (the ones used in the case of *PETS09-S2L2*, *Soccer* and *Town Center (Body)* sequences), the proposed tracker provides a high MOTA value when γ_{CF} is in the interval of $[0.5, 0.6]$.

Number of Key Samples: We analyze the effect of the number of key samples retained on MOTA using the *PETS09-S2L1* sequence. The appearance model (SDC, SGM, and PGM) is updated online at an update rate R_u of 10. Figure 4.11 shows the performance of the proposed tracker when the number of key samples retained is varied. We choose the number of retained key samples to be 20 at which the highest MOTA performance is exhibited, as seen from Figure 4.11.

Key Sample Selection: To demonstrate the strength of the proposed sample selection scheme, we examine the performance of the proposed tracking scheme by varying the SDC similarity threshold, s_0, from 0 to 1.5 in increments of 0.1. Figure 4.12 shows the performance of the proposed scheme at different SDC tracker similarity threshold values. When $s_0 = 1$, the proposed tracker exhibits the best performance in terms of MOTA. If $0 \leq s_0 < 1$, the performance is not as good in view of the fact that only a few or none of the key samples are rejected, and hence, occluded samples are likely to be selected. When $s_0 > 1.2$, the proposed tracker performs worse than that at $s_0 = 1$, since a large number of key samples are rejected. As such, we choose $s_0 = 1.0$ for all the experiments.

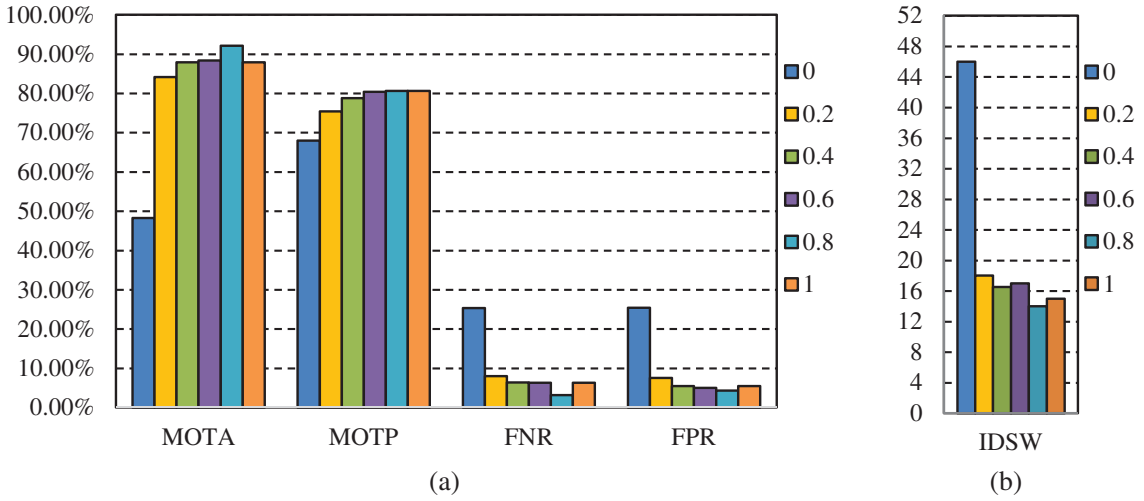


Figure 4.10: Performance of the proposed method on the *PETS09-S2L1* sequence for different values of the collaborative factor γ_{CF}.

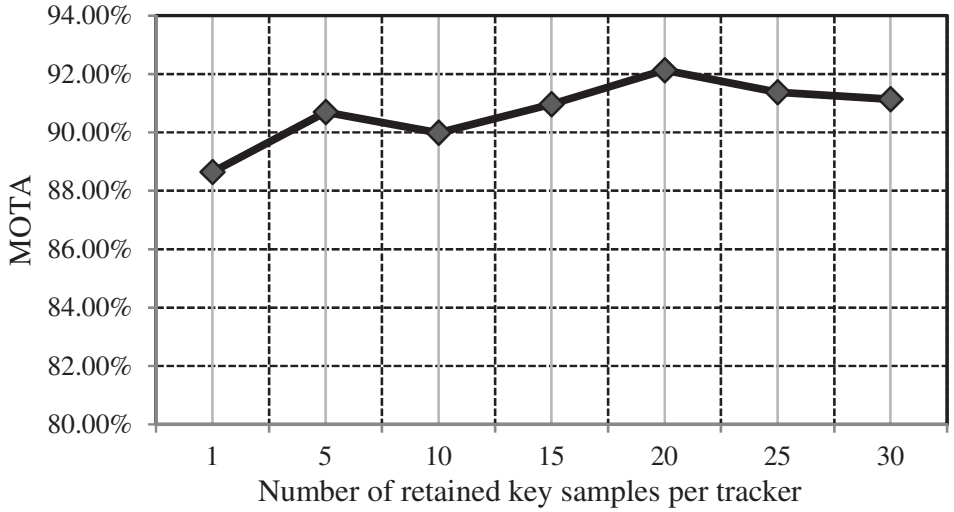


Figure 4.11: MOTA vs. number of retained key samples for the proposed tracker on the *PETS09-S2L1* sequence.

Effect of Tracker Re-detection: We analyze the effect of using the re-detection module on multi-object tracking. Figure 4.13 shows that the proposed method with tracker re-detection scheme achieves slightly lower FNR and FPR than that obtained without using the tracker re-detection scheme, while maintaining approximately the same performance in terms of MOTA and MOTP values. The tracker re-detection scheme aims to reduce the number of identity switches and maintains long trajectories, without reducing the tracking performance.

Generative Appearance Models: We study the tracking performance of the proposed method by using several types of generative models to solve the data association problem in (4.23). These generative models are (1) SGM, as outlined in Section 4.2.2, which is based on local patch features (by substituting in (4.23) by $G = G_{SGM}$); (2) 2DPCA generative model, as proposed in Section 4.2.2, which is based on holistic features (by substituting in (4.23) by $G = G_{PGM}$); (3) combination of SGM and 2DPCA generative models as mentioned in Section 4.2.4; (4) principal component analysis (PCA)[1] generative model (instead of using the 2DPCA generative model); and (5) combination of SGM and PCA generative models.

The main differences between 2DPCA versus PCA are as follows. The covariance matrix in the case of 2DPCA can be computed directly from the image samples in 2D matrices rather than 1D vectors as in the case of PCA [94, 120]. The complexity for computing the covariance matrix using a 2DPCA-based appearance model is $\mathcal{O}(mn^2N)$, whereas the corresponding complexity using a PCA-based appearance model is $\mathcal{O}(m^2n^2N)$, when a set of N image samples, each of size $m \times n$ pixels, is used. Further, it may be pointed out that 2DPCA encodes the relationship among neighboring rows in a given set of image samples [120]. Such a relationship should have a positive effect on the tracking performance.

Table 4.1 shows the results on the seven sequences. Overall, the proposed scheme with SGM in conjunction with 2DPCA performs better than that by using SGM with

[1]The function `pcaApply` from toolbox [97] has been used to calculate the PCA.

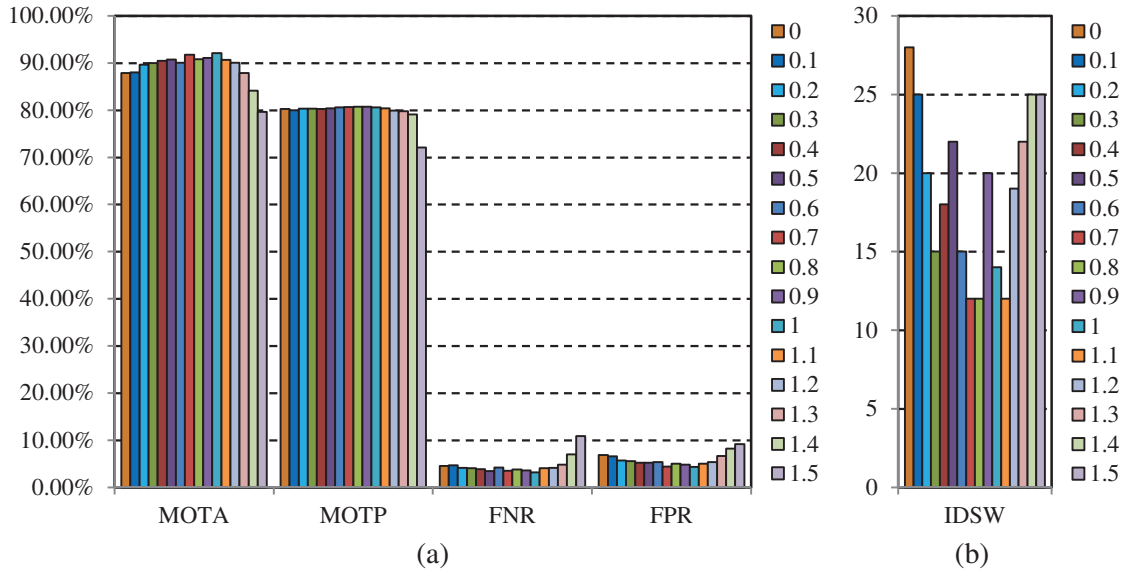


Figure 4.12: Performance of the proposed tracking scheme with respect to the SDC similarity threshold, s_0, using the *PETS09-S2L1* sequence.

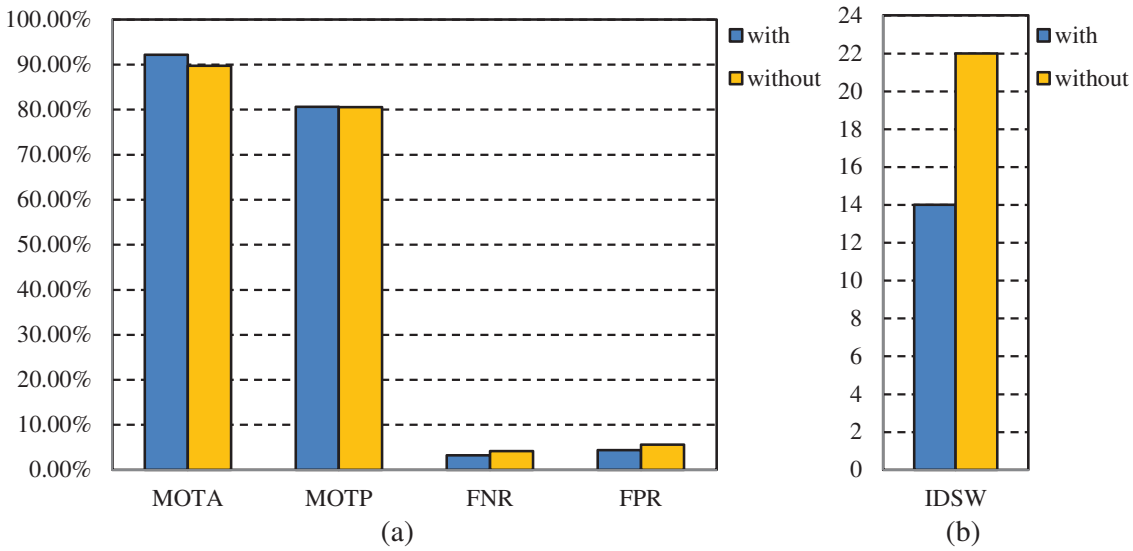


Figure 4.13: Performance of the proposed method with and without tracker re-detection on the *PETS09-S2L1* sequence.

PCA. In most sequences, the method of using SGM with 2DPCA or SGM with PCA performs better than that using only SGM. On a machine with 2.9 GHz CPU, the average tracking time per frame (over all the seven sequences without counting the time for object detection) for the proposed tracker with SGM and 2DPCA is 2.88 s whereas the corresponding time in the case of SGM and PCA is 2.90 s. Hence, this improvement in the performance of the proposed tracker is achieved without loss in speed.

4.3.4 Performance Comparison

In this section, we evaluate the performance of the proposed algorithm with two online multi-object tracking methods in [121, 122] using the seven challenging sequences described in Section 4.3.1. Table 4.2 shows the performance of these two methods (using the original source code) along with that of the proposed tracker in terms of the various CLEAR MOT metrics. In addition, the performance of the proposed scheme is compared with the reported results of state-of-the-art online multi-object tracking methods [63, 71, 88, 123–126] using the sequences considered in these papers.

On the *PETS09-S2L1* and *PETS09-S2L2* sequences, the proposed scheme provides the second highest MOTA values. It also offers the highest and second highest MOTP values on the *PETS09-S2L1* and *PETS09-S2L2* sequences, respectively. This can be attributed to the proposed update mechanism, and the inactive or on-hold states of the tracker.

For the *Soccer* sequence, the proposed scheme performs better than the methods in [121, 122] despite fast camera motion and the presence of similar objects in the scenes. For the *UCF-PL* sequence, the MOTA value of the proposed method is higher than that of the methods in [63, 121, 122], using the same detector as in [63]. On the other hand, the MOTP value of the proposed technique is close to that of [63]. In addition, the proposed method has lower values for FNR and FPR than the methods in [63, 121, 122] do.

For the *Town Center* dataset, the proposed scheme is first evaluated to track the full body of pedestrians. In this case, the proposed scheme yields the second highest MOTP, FNR and FPR values compared to the methods in [63, 71, 88, 121, 122]. Next, the proposed scheme is evaluated on tracking the heads of pedestrians from the same dataset. The head regions in this sequence are less occluded than the full body, although the head detector has higher FPR than the full-body detector. As shown in Table 4.2, the proposed method performs well against other approaches [88, 121, 122, 126] in terms of MOTA. For the *Urban* and *Sunny* sequences from *LISA 2010* dataset, the proposed scheme provides a better performance than that provided by the methods in [121, 122] for tracking multiple vehicles on-road.

We note that the proposed scheme uses grayscale images as features, whereas the methods in [63, 71, 124] are based on the color or gradient information of the targets. In addition, the proposed scheme does not require the detector confidence density or a gate function in the data association step as in [5, 124], where the gate function provides higher weight for detections located in the direction of motion of the target.

4.4 Summary

In this chapter, we have presented a robust collaborative model that enhances the interaction between a pre-trained object detector and a number of single-object online trackers in the particle filter framework. The proposed scheme is based on incorporating the associated detections with the motion model, in addition to the likelihood function providing different weights for the propagated and the newly created particles sampled from the associated detections, offering a reduction on the effect of the detector errors on the tracking process. We have exploited sparse representation and 2DPCA to construct diverse features that maximize the appearance variation among the trackers. Furthermore, we have presented a conservative sample selection scheme to update the appearance model of every tracker. Experimental results on benchmark datasets have shown that the proposed scheme outperforms state-of-the-art multi-object tracking methods in most of the cases.

Table 4.1: Performance of the proposed scheme using different generative models.

Sequence	Generative model	MOTA	MOTP	FNR	FPR	IDSW
	SGM	89.08%	79.89%	5.11%	5.42%	17
	PCA	89.86%	79.97%	5.04%	4.76%	16
PETS09-S2L1	SGM + PCA	90.12%	80.55%	5.34%	**4.30%**	**13**
	2DPCA	89.81%	79.82%	5.40%	4.41%	20
	Proposed	**92.13%**	**80.62%**	**3.19%**	4.33%	14
	SGM	36.43%	71.19%	39.38%	26.31%	263
	PCA	45.69%	71.74%	35.92%	20.25%	**218**
PETS09-S2L2	SGM + PCA	44.35%	71.54%	36.04%	21.30%	237
	2DPCA	46.06%	**71.77%**	36.59%	**19.33%**	221
	Proposed	**46.88%**	71.66%	**34.92%**	19.43%	258
	SGM	67.36%	70.28%	16.72%	14.49%	45
	PCA	70.33%	70.64%	18.85%	10.28%	38
Soccer	SGM + PCA	70.21%	**70.99%**	18.20%	11.03%	**36**
	2DPCA	71.13%	70.73%	17.00%	10.66%	49
	Proposed	**73.54%**	70.77%	**16.20%**	**9.45%**	38
	SGM	82.30%	71.84%	10.77%	6.27%	16
	PCA	82.14%	71.88%	10.44%	6.56%	21
UCF-PL	SGM + PCA	83.29%	71.81%	10.64%	**5.40%**	16
	2DPCA	81.89%	71.75%	11.47%	5.90%	18
	Proposed	**85.02%**	**71.89%**	**8.70%**	5.65%	**15**
	SGM	69.41%	73.82%	**17.08%**	12.81%	444
	PCA	70.19%	73.83%	18.18%	11.08%	351
Town Center (Body)	SGM + PCA	69.83%	73.89%	19.29%	10.37%	**320**
	2DPCA	**71.24%**	**74.02%**	18.02%	10.21%	337
	Proposed	70.16%	73.93%	19.35%	**9.95%**	342
	SGM	70.32%	68.86%	14.96%	14.48%	164
	PCA	72.15%	68.71%	14.25%	13.37%	163
Town Center (Head)	SGM + PCA	69.37%	68.78%	15.62%	14.77%	166
	2DPCA	70.43%	68.82%	15.06%	14.29%	**158**
	Proposed	**74.54%**	**69.15%**	**13.02%**	**12.21%**	**158**
	SGM	100.00%	82.67%	0.00%	0.00%	0
	PCA	100.00%	82.68%	0.00%	0.00%	0
LISA10 Urban	SGM + PCA	100.00%	82.68%	0.00%	0.00%	0
	2DPCA	100.00%	82.68%	0.00%	0.00%	0
	Proposed	100.00%	82.68%	0.00%	0.00%	0
	SGM	97.22%	78.28%	0.78%	1.98%	0
	PCA	97.22%	78.28%	0.78%	1.98%	0
LISA10 Sunny	SGM + PCA	97.22%	78.28%	0.78%	1.98%	0
	2DPCA	97.22%	78.28%	0.78%	1.98%	0
	Proposed	97.22%	78.28%	0.78%	1.98%	0
	SGM	76.51%	74.60%	13.10%	10.22%	-
	PCA	78.45%	74.72%	12.93%	8.53%	-
Average	SGM + PCA	78.05%	74.81%	13.24%	8.64%	-
	2DPCA	78.47%	74.73%	13.04%	8.35%	-
	Proposed	**79.94%**	**74.87%**	**12.02%**	**7.87%**	-

Note: The best and the second best results on each dataset are shown in boldface and underscored, respectively. The proposed method is SGM + 2DPCA.

Table 4.2: Performance measures of CLEAR MOT metrics.

Sequence	Method	MOTA	MOTP	FNR	FPR	IDSW
PETS09-S2L1	Proposed	92.13%	**80.62%**	**3.19%**	**4.33%**	14
	Yoon *et al.* [121]*	66.64%	57.46%	17.99%	15.14%	34
	Bao and Yoon [122]*	89.94%	79.34%	4.83%	4.73%	23
	Zhang *et al.* [71]	**93.27%**	68.17%	-	-	19
	Zhou *et al.* [123]	87.21%	58.47%	-	-	-
	Breitenstein *et al.* [124]	79.70%	56.30%	-	-	-
	Gerónimo *et al.* [125]	51.10%	75.00%	45.20%	-	**0**
PETS09-S2L2	Proposed	46.88%	71.66%	**34.92%**	19.43%	258
	Yoon *et al.* [121]*	26.85%	47.99%	51.27%	28.86%	218
	Bao and Yoon [122]*	45.98%	**71.77%**	35.73%	**19.06%**	325
	Zhang *et al.* [71]	**66.72%**	58.21%	-	-	**215**
Soccer	Proposed	**73.54%**	**70.77%**	**16.20%**	**9.45%**	38
	Yoon *et al.* [121]*	29.99%	53.77%	52.89%	26.19%	**10**
	Bao and Yoon [122]*	54.25%	69.26%	35.45%	12.64%	24
UCF-PL	Proposed	**85.02%**	71.89%	**8.70%**	**5.65%**	**15**
	Yoon *et al.* [121]*	29.50%	45.33%	38.04%	33.95%	**15**
	Bao and Yoon [122]*	82.84%	73.33%	10.31%	6.49%	**15**
	Shu *et al.* [63]	79.30%	**74.10%**	18.30%	8.70%	-
Town Center (Body)	Proposed	70.16%	73.93%	19.35%	9.95%	342
	Yoon *et al.* [121]*	62.93%	48.66%	20.00%	17.14%	330
	Bao and Yoon [122]*	**79.07%**	73.46%	**11.19%**	**9.44%**	**307**
	Benfold and Reid [88]	61.30%	**80.30%**	21.00%	18.00%	-
	Zhang *et al.* [71]	73.61%	68.75%	-	-	421
	Shu *et al.* [63]	72.90%	71.30%	-	-	-
Town Center (Head)	Proposed	**74.54%**	69.15%	**13.02%**	12.21%	158
	Yoon *et al.* [121]*	73.90%	**70.16%**	17.23%	**9.49%**	**126**
	Bao and Yoon [122]*	70.65%	69.97%	16.31%	13.07%	320
	Poiesi *et al.* [126]	54.60%	63.70%	23.80%	21.70%	285
	Benfold and Reid [88]	45.40%	50.80%	29.00%	26.20%	-
LISA10 Urban	Proposed	**100.00%**	**82.68%**	**0.00%**	**0.00%**	**0**
	Yoon *et al.* [121]*	99.33%	81.98%	0.33%	0.33%	**0**
	Bao and Yoon [122]*	98.33%	82.52%	1.67%	**0.00%**	**0**
LISA10 Sunny	Proposed	**97.22%**	**78.28%**	**0.78%**	1.98%	**0**
	Yoon *et al.* [121]*	92.89%	77.20%	6.89%	**0.24%**	**0**
	Bao and Yoon [122]*	97.00%	77.83%	2.67%	0.34%	**0**

Note: $*$ denotes the results obtained by utilizing the code provided by the authors of the paper, where the detection results and GT annotations that have been used with the proposed scheme are used. The best and the second best results on each dataset are represented in boldface and underscored, respectively.

Chapter 5

Conclusion

5.1 Concluding Remarks

Multi-object detection and tracking has many promising applications in the field of computer vision, such as human activity recognition, human computer interaction, crowd scene analysis, video surveillance, sports video analysis, autonomous vehicles navigation, driver assistance systems, and traffic management. In this thesis, a novel object detection technique using the two-dimensional discrete Fourier or cosine transform and a detection-based online multi-object tracking technique have been developed.

In the first part of the thesis, a new vehicle detection scheme using transform-domain 2DHOG features has been proposed. This scheme is based on extracting from the input image the transform domain based features, referred to as the transform-domain 2DHOG (TD2DHOG) features. It has been shown that the TD2DHOG features so obtained at an original resolution and a downsampled version of the same image are approximately the same within a multiplicative factor. This property has been then utilized in developing a scheme for the detection of vehicles of various resolutions using a single classifier rather than multiple resolution-specific classifiers. Experimental results on three vehicle detection datasets, namely, *UIUC car detection dataset* [83],

the *USC multi-view car detection dataset* [28], and the *LISA 2010 dataset* [84], have shown that the use of the single classifier in the proposed detection scheme reduces the training cost by at least 44.63% and the storage requirement by at least 50.00% over the use of a classifier pyramid, yet provides a detection accuracy similar to that obtained using TD2DHOG features with a classifier pyramid. In addition, the proposed method provides a detection accuracy that is similar to or even better than that provided by the state-of-the-art techniques. Experimental results have shown that the proposed scheme works well under several challenging conditions such as variation in scale, appearance, view of the objects, as well as partial occlusion, and changes in illumination conditions.

In the second part of the thesis, a collaborative model between a pre-trained object detector and a number of single-object online trackers has been presented and used to develop a detection-based online multi-object tracking scheme. For each frame, an association a detection and a tracker has been constructed. For each tracker, a motion model that incorporates the associated detections with object dynamics, and a likelihood function that provides different weights for the propagated particles and the newly created ones from the associated detections have been proposed. An effective sample selection scheme has been introduced to update the appearance model of a given tracker. It has been shown that the proposed collaborative model, which weights differently the propagated and the newly created particles, improved the multiple object tracking accuracy (MOTA), false negative rate (FNR) and false positive rate (FPR) of the proposed tracker by 4.63%, 49.48% and 20.56%, respectively, over that of the tracker which weights the two sets of particles equally. Experimental results on seven challenging sequences, namely, the *PETS09-S2L1, PETS09-S2L2* [87], *UCF Parking Lot (UCF-PL)* dataset [63], *Soccer* dataset [62], *Town Center* dataset [88], and *Urban* as well as *Sunny* sequences from *LISA 2010* dataset [84], have shown that the proposed scheme generally outperforms the state-of-the-art methods.

The study undertaken in this thesis has shown that two-dimensional transform-domain based features can be used to design an object detector that not only reduces the storage and training costs, but also offers a high detection accuracy; further, the effect of detection errors on the tracking process can be alivated by using a collaborative model that depends on the propagated particles and the newly-created ones from the associated detections. Thus, this study may enable further work in the design of transform-domain based features that are able to tackle the challenges of orientation, aspect-ratio and scale change in object detection and tracking problems, as well as in building a collaborative model for multi-object tracking that includes more challenging conditions such as heavy-occlusion and various motion patterns.

5.2 Scope for Future Investigation

The research work presented in this thesis can be extended in a number of ways. The spatial domain two-dimensional HOG features can be replaced by other features such as gradient magnitude or color features prior to taking the DFT or DCT transform. The transform used itself could be other transforms such as wavelet, curvelet or contourlet. Depending on the spatial domain features and the transform used the relationship between the transformed features at two different resolutions could be investigated. Unlike the work of this thesis in which the detection is carried out using the frequency domain features, the detection process could be investigated using some suitable spatial domain features. In this thesis, tracking algorithms have been developed using spatial domain features. The use of frequency domain features could also be investigated for the purpose of multi-object tracking. Finally, instead of extracting the spatial domain features individually for each frame, a tracking scheme could be developed in which the motion information is used to determine the features of the succeeding frames.

References

[1] T. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *J. Comput. Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[2] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 28, no. 5, pp. 694–711, 2006.

[3] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. on Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, 2011.

[4] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 34, no. 4, pp. 743–761, 2012.

[5] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2009, pp. 1515–1522.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2005, pp. 886–893.

[7] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. Sixth Int. Conf. on Comput. Vision (ICCV)*, 1998, pp. 555–562.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. on Comput. Vision (ICCV)*, 1999, pp. 1150–1157.

[9] ——, "Distinctive image features from scale-invariant keypoints," *Int. J. of Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. of the 7th European Conf. on Comput. Vision (ECCV)*, 2002, pp. 128–142.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Comput. Vision and Image Understanding (CVIU)*, vol. 110, pp. 346–359, 2008.

[12] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. of Comput. Vision*, vol. 77, no. 1, pp. 259–289, 2008.

[13] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.

[14] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE Int. Conf. on Comput. Vision (ICCV)*, 2009, pp. 32–39.

[15] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D voxel patterns for object category recognition," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, 2015, pp. 1903–1911.

[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, vol. 1, June 2001, pp. 511–518.

[17] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proc. Int. Conf. on Image Processing (ICIP)*, vol. 1, 2002, pp. 900–903.

[18] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 23, no. 4, pp. 349–361, 2001.

[19] R. R. Cabrera, T. Tuytelaars, and L. V. Gool, "Efficient multi-camera detection, tracking, and identification using a shared set of haar-features," in *Proc. IEEE Conf. On Comput. Vision And Pattern Recogn. (CVPR)*, 2011, pp. 65–71.

[20] N. Dalal, "Finding people in images and videos," Ph.D. dissertation, Institut National Polytechnique de Grenoble, July 2006.

[21] B.-F. Wu, C.-C. Kao, C.-L. Jen, Y.-F. Li, Y.-H. Chen, and J.-H. Juang, "A relative-discriminative-histogram-of-oriented-gradients-based particle filter approach to vehicle occlusion handling and tracking," *IEEE Trans. on Industrial Electronics*, vol. 61, pp. 4228–4237, 2014.

[22] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, 2008, pp. 1–8.

[23] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2009, pp. 91.1–91.11.

[24] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[25] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, vol. 2, 2006, pp. 1491–1498.

[26] I. Laptev, "Improvements of object detection using boosted histograms," in *Proc. of the British Machine Vision Conf. (BMVC)*, vol. III, 2006, pp. 949–958.

[27] ——, "Improving object detection with boosted histograms," *Image and Vision Computing*, vol. 27, no. 5, pp. 535–544, 2009.

[28] C.-H. Kuo and R. Nevatia, "Robust multi-view car detection using unsupervised subcategorization," in *Proc. IEEE Workshop on Appl. of Comput. Vision (WACV)*, 2009, pp. 1–8.

[29] F. Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, vol. 1, 2005, pp. 829–836.

[30] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. Computational Learning Theory: Eurocolt*, 1995, pp. 23–37.

[31] C. Zhang and P. Viola, "Multiple-instance pruning for learning efficient cascade detectors," in *Proc. Neural Information Processing Systems (NIPS)*, 2007.

[32] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2010, pp. 68.1–68.11.

[33] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 36, no. 8, pp. 1532–1545, 2014.

[34] R. Benenson, M. Mathias, R. Timofte, and L. V. Gool, "Pedestrian detection at 100 frames per second," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, 2012, pp. 2903–2910.

[35] E. Ohn-Bar and M. M. Trivedi, "Learning to detect vehicles by clustering appearance patterns," *IEEE Trans. on Intell. Transp. Syst. (ITS)*, vol. 16, no. 5, pp. 2511–2521, 2015.

[36] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg, "Part-based feature synthesis for human detection," in *Proc. European Conf. on Comput. vision (ECCV): Part I*, 2010, pp. 127–142.

[37] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *Proc. European Conf. On Comput. Vision (ECCV)*, May 2004, pp. 69–82.

[38] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. of Comput. Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[39] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, "Latent hierarchical structural learning for object detection," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, June 2010, pp. 1062–1069.

[40] G. Duan, H. Ai, and S. Lao, "A structural filter approach to human detection," in *Proc. of the 11th European Conf. on Comput. vision (ECCV): Part VI*, 2010, pp. 238–251.

[41] A. Takeuchi, S. Mita, and D. McAllester, "On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, 2010, pp. 1014–1021.

[42] S. Sivaraman and M. M. Trivedi, "Vehicle detection by independent parts for urban driver assistance," *IEEE Trans. on Intell. Transp. Syst. (ITS)*, vol. 14, no. 4, pp. 1597–1608, 2013.

[43] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. on Computers*, vol. 22, no. 1, pp. 67–92, 1973.

[44] B. Li, T. Wu, and S.-C. Zhu, "Integrating context and occlusion for car detection by hierarchical and-or model," in *Proc. European Conf. on Comput. Vision (ECCV)*, 2014, pp. 652–667.

[45] C. Wang, Y. Fang, H. Zhao, C. Guo, S. Mita, and H. Zha, "Probabilistic inference for occluded and multiview on-road vehicle detection," *IEEE Trans. on Intell. Transp. Syst. (ITS)*, vol. 17, no. 1, 2016.

[46] B. Pepikj, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, 2013, pp. 3286–3293.

[47] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 37, no. 10, pp. 2071–2084, 2015.

[48] C. Dubout and F. Fleuret, "Exact acceleration of linear object detectors," in *Proc. European Conf. on Comput. Vision (ECCV)*, 2012, pp. 301–311.

[49] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, June 2011, pp. 1713–1720.

[50] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2011, pp. 76.1–76.12.

[51] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn, "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2011, pp. 120–127.

[52] W. Brendel, M. R. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 1273–1280.

[53] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Proc. Comput. Vis. Pattern Recognit.*, 2011, pp. 1265–1272.

[54] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking multiple people under global appearance constraints," in *Proc. Int. Conf. Comput. Vis. (ICCV)*.

[55] A. R. Zamir, A. Dehghan, and M. Shah, "GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs," in *Proc. European Conf. Comput. Vis. (ECCV)*, 2012, pp. 343–356.

[56] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 1926–1933.

[57] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 1948–1955.

[58] H. Izadinia, I. Saleemi, W. Li, and M. Shah, "$(MP)^2T$: Multiple people multiple parts tracker," in *Proc. European Conf. Comput. Vis.*, 2012, pp. 100–114.

[59] A. A. Butt and R. T. Collins, "Multiple target tracking using frame triplets," in *Proc. Asian Conf. on Comput. Vis.*, 2012, pp. 163–176.

[60] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking, 2015," *arXiv:1504.01942*, April 2015.

[61] K. Okuma, A. Taleghani, N. D. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. European Conf. Comput. Vis. (ECCV)*, 2004, pp. 28–39.

[62] Y. Wu, X. Tong, Y. Zhang, and H. Lu, "Boosted interactively distributed particle filter for automatic multi-object tracking," in *Proc. Int. Conf. on Image Process. (ICIP)*, 2008, pp. 1844–1847.

[63] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 1815–1821.

[64] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multimodality through mixture tracking," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2003, pp. 1110–1116.

[65] Y. Jin and F. Mokhtarian, "Variational particle filter for multi-object tracking," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2007, pp. 1–8.

[66] S. Duffner and J. Odobez, "Track creation and deletion framework for long-term online multi-face tracking," *IEEE Trans. on Image Process.*, vol. 22, no. 1, pp. 272–285, 2013.

[67] E. Maggio, M. Taj, and A. Cavallaro, "Efficient multitarget visual tracking using random finite sets," *IEEE Trans. on Circuits and Syst. for Video Technology*, vol. 18, no. 8, pp. 1016–1027, 2008.

[68] V. Eiselein, D. Arp, M. Patzold, and T. Sikora, "Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors," in *Proc. IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance*, 2012, pp. 325–330.

[69] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. on Aerosp. and Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, 2003.

[70] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro, "Particle PHD filtering for multi-target visual tracking," in *Proc. Int. Conf. on Acoust., Speech, and Signal Process.*, vol. 1, 2007, pp. I–1101–I–1104.

[71] J. Zhang, L. L. Presti, and S. Sclaroff, "Online multi-person tracking by tracker hierarchy," in *Proc. IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance*, 2012, pp. 379–385.

[72] M. Yang, F. Lv, W. Xu, and Y. Gong, "Detection driven adaptive multi-cue integration for multiple human tracking," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1554–1561.

[73] A. Schumann, M. Bäuml, and R. Stiefelhagen, "Person tracking-by-detection with efficient selection of part-detectors," in *IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance*, 2013, pp. 43–50.

[74] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear and non-gaussian bayesian state estimation," *IEE Proc. F (Radar and Signal Process.)*, vol. 140, no. 2, pp. 107–113, 1993.

[75] Y. Jinxia, T. Yongli, X. Jingmin, and Z. Qian, "Research on particle filter based on an improved hybrid proposal distribution with adaptive parameter optimization," in *Proc. Int. Conf. on Intell. Computation Technol. and Automation*, 2012, pp. 406–409.

[76] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Proc. Comput. Vis. Pattern Recognit.*, 2001, pp. 786–793.

[77] Y. Huang and P. M. Djuric, "A hybrid importance function for particle filtering," *IEEE Signal Process. Letters*, vol. 11, no. 3, pp. 404–406, 2004.

[78] S. Santhoshkumar, S. Karthikeyan, and B. S. Manjunath, "Robust multiple object tracking by detection with interacting markov chain Monte Carlo," in *Proc. Int. Conf. on Image Process.*, 2013.

[79] H. Han, Y.-S. Ding, K.-R. Hao, and X. Liang, "An evolutionary particle filter with the immune genetic algorithm for intelligent video target tracking," *Comput. and Mathematics with Applicat.*, vol. 62, no. 7, pp. 2685–2695, 2011.

[80] A. Doucet, N. D. Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*, ser. Statistics for Eng. and Inf. Sci. Springer New York, 2001.

[81] M. A. Naiel, M. O. Ahmad, and M. N. S. Swamy, "A vehicle detection scheme based on two-dimensional HOG features in the DFT and DCT domains," *IEEE Trans. on Intell. Transp. Syst. (ITS)*, 2016, under review.

[82] ——, "Vehicle detection using TD2DHOG features," in *Proc. New Circuits and Syst. Conf.*, 2014, pp. 389–392.

[83] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 26, no. 11, pp. 1475–1490, 2004, http://cogcomp.cs.illinois.edu/Data/Car/, last retrieved, December 13, 2016.

[84] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. on Intell. Transp. Syst. (ITS)*, vol. 11, no. 2, pp. 267–276, 2010.

[85] M. A. Naiel, M. O. Ahmad, M. N. S. Swamy, J. Lim, and M.-H. Yang, "Online multi-object tracking via robust collaborative model and sample selection," *Computer Vision and Image Understanding*, vol. 154, pp. 94–107, Jan. 2017, http://dx.doi.org/10.1016/j.cviu.2016.07.003.

[86] M. A. Naiel, M. O. Ahmad, M. N. S. Swamy, Y. Wu, and M.-H. Yang, "Online multi-person tracking via robust collaborative model," in *Proc. Int. Conf. on Image Process. (ICIP)*, 2014, pp. 431–435.

[87] J. Ferryman, in *Proc. IEEE Workshop Performance Evaluation of Tracking and Surveillance*, 2009.

[88] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 3457–3464.

[89] D. L. Ruderman, "The statistics of natural images," *Network: Computation in Neural Systems*, vol. 5, pp. 517–548, 1994.

[90] J. Huang and D. Mumford, "Statistics of natural images and models," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, vol. 1, 1999, pp. 541–547.

[91] J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT), with Audio Applications — Second Edition*. W3K Publishing, 2007, http://ccrma.stanford.edu/~jos/mdft/, last retrieved, December 13, 2016.

[92] G. Bi and S. K. Mitra, "Sampling rate conversion in the frequency domain [dsp tips and tricks]," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 140–144, 2011.

[93] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. on Computers*, vol. C-23, no. 1, pp. 90–93, 1974.

[94] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 26, no. 1, pp. 131–137, 2004.

[95] S. Maji, A. C. Berg, and J. Malik, "Efficient classification for additive kernel SVMs," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 35, no. 1, pp. 66–77, 2013.

[96] R. Appel, T. Fuchs, P. Dollár, and P. Perona, "Quickly boosting decision trees - pruning underachieving features early," in *Proc. Inter. Conf. on Machine Learning (ICML)*, 2013, pp. 594–602.

[97] P. Dollár, "Piotr's Image and Video Matlab Toolbox (PMT)," http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html, last retrieved, December 12, 2016.

[98] A. Gepperth, S. Rebhan, S. Hasler, and J. Fritsch, "Biased competition in visual processing hierarchies: A learning approach using multiple cues," *Cognitive computation*, vol. 3, no. 1, pp. 146–166, 2011.

[99] S. Bileschi, "StreetScenes: Towards scene understanding in still images," Ph.D. dissertation, Massachusetts Institute of Technology, 2006, CBCL dataset link: http://cbcl.mit.edu/software-datasets/streetscenes, last retrieved, December 13, 2016.

[100] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, 2012, pp. 3354–3361.

[101] J. Mutch and D. G. Lowe, "Object class recognition and localization using sparse features with limited receptive fields," *Int. J. of Comput. Vision*, vol. 80, no. 1, pp. 45–57, 2008.

[102] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, 2008, pp. 1–8.

[103] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, 2009, pp. 1022–1029.

[104] J. Wu, N. Liu, C. Geyer, and J. M. Rehg, "C^4: A real-time object detection framework," *IEEE Trans. on Image Processing*, vol. 22, no. 10, pp. 4096–4107, 2013.

[105] M. Everingham, L. V. Gool, C. K. Williams, J.Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[106] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge 2007 (VOC2007) results," http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html, last retrieved, December 13, 2016.

[107] B. Wu and R. Nevatia, "Cluster boosted tree classifier for multi-view, multi-pose object detection," in *Proc. Int. Conf. on Comput. Vision (ICCV)*, 2007, pp. 1–8.

[108] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[109] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[110] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 1838–1845.

[111] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 31, no. 2, pp. 210–227, 2009.

[112] T. Wang, I. Y. H. Gu, and P. Shi, "Object tracking using incremental 2D-PCA learning and ML estimation," in *Proc. Int. Conf. on Acoust., Speech, and Signal Process.*, vol. 1, 2007, pp. 933–936.

[113] C.-H. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *Proc. Comput. Vis. Pattern Recognit.*, 2010, pp. 685–692.

[114] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[115] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *Proc. Comput. Vis. Pattern Recognit.*, June 2012, pp. 1918–1925.

[116] http://iris.usc.edu/people/yangbo/downloads.html, last retrieved, December 13, 2016.

[117] http://research.milanton.de/data.html, last retrieved, December 13, 2016.

[118] http://crcv.ucf.edu/data/ParkingLOT/index.php, last retrieved, December 13, 2016.

[119] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *J. on Image and Video Process.*, pp. 1–10, 2008.

[120] D. Zhanga and Z.-H. Zhoua, "(2D)^{2}PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1-3, pp. 224–231, 2005.

[121] J. H. Yoon, M.-H. Yang, J. Lim, and K. J. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *Proc. IEEE Winter Conf. on Applicat. of Comput. Vis.*, 2015, pp. 33–40.

[122] S. H. Bae and K. J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proc. Comput. Vis. Pattern Recognit.*, 2014, pp. 1218–1225.

[123] X. Zhou, Y. Li, B. He, and T. Bai, "GM-PHD-based multi-target visual tracking using entropy distribution and game theory," *IEEE Trans. on Ind. Informat.*, vol. 10, no. 2, pp. 1064–1076, 2014.

[124] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 33, no. 9, pp. 1820–1833, 2011.

[125] D. G. Gomez, F. Lerasle, and A. M. L. Peña, "State-driven particle filter for multi-person tracking," in *Proc. Advanced Concepts for Intell. Vis. Syst.*, ser. Lecture Notes in Comput. Sci., vol. 7517, 2012, pp. 467–478.

[126] F. Poiesi, R. Mazzon, and A. Cavallaro, "Multi-target tracking on confidence maps: An application to people tracking," *Comput. Vis. and Image Under.*, vol. 117, no. 10, pp. 1257 – 1272, 2013.