Visual Tracking Algorithms using Different Object Representation Schemes

Shreyamsha Kumar Bidare Kantharajappa

A Thesis

In the Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

June, 2019

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Shreyamsha Kumar Bidare Kantharajappa

Entitled: Visual Tracking Algorithms using Different Object Representation Schemes

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Rene Witte

_____ External Examiner
Dr. Panajotis Agathoklis

_____ External to Program
Dr. Terry Fancott

_____ Examiner
Dr. Hassan Rivaz

_____ Examiner
Dr. Wei-Ping Zhu

_____ Thesis Co-Supervisor
Dr. M. Omair Ahmad

_____ Thesis Co-Supervisor
Dr. M.N.S. Swamy

Approved by _____
Dr. Rastko R. Selmic, Graduate Program Director

August 2, 2019

_____
Dr. Amir Asif, Dean
Gina Cody School of Engineering & Computer Science

Abstract

Visual Tracking Algorithms using Different Object Representation Schemes

Shreyamsha Kumar Bidare Kantharajappa, Ph.D.

Concordia University, 2019

Visual tracking, being one of the fundamental, most important and challenging areas in computer vision, has attracted much attention in the research community during the past decade due to its broad range of real-life applications. Even after three decades of research, it still remains a challenging problem in view of the complexities involved in the target searching due to intrinsic and extrinsic appearance variations of the object. The existing trackers fail to track the object when there are considerable amount of object appearance variations and when the object undergoes severe occlusion, scale change, out-of-plane rotation, motion blur, fast motion, in-plane rotation, out-of-view and illumination variation either individually or simultaneously. In order to have a reliable and improved tracking performance, the appearance variations should be handled carefully such that the appearance model should adapt to the intrinsic appearance variations and be robust enough for extrinsic appearance variations. The objective of this thesis is to develop visual object tracking algorithms by addressing the deficiencies of the existing algorithms to enhance the tracking performance by investigating the use of different object representation schemes to model the object appearance and then devising mechanisms to update the observation models.

A tracking algorithm based on the global appearance model using robust coding and its collaboration with a local model is proposed. The global PCA subspace is used to model the global appearance of the object, and the optimum PCA basis coefficients and the global weight matrix are estimated by developing an iteratively reweighted robust coding (IRRC) technique. This global model is collaborated with the local

model to exploit their individual merits. Global and local robust coding distances are introduced to find the candidate sample having similar appearance as that of the reconstructed sample from the subspace, and these distances are used to define the observation likelihood. A robust occlusion map generation scheme and a mechanism to update both the global and local observation models are developed. Quantitative and qualitative performance evaluations on OTB-50 and VOT2016, two popular benchmark datasets, demonstrate that the proposed algorithm with histogram of oriented gradient (HOG) features generally performs better than the state-of-the-art methods considered do. In spite of its good performance, there is a need to improve the tracking performance in some of the challenging attributes of OTB-50 and VOT2016.

A second tracking algorithm is developed to provide an improved performance in situations for the above mentioned challenging attributes. The algorithms is designed based on a structural local 2DDCT sparse appearance model and an occlusion handling mechanism. In a structural local 2DDCT sparse appearance model, the energy compaction property of the transform is exploited to reduce the size of the dictionary as well as that of the candidate samples in the object representation so that the computational cost of the l_1-minimization used could be reduced. This strategy is in contrast to the existing models that use raw pixels. A holistic image reconstruction procedure is presented from the overlapped local patches that are obtained from the dictionary and the sparse codes, and then the reconstructed holistic image is used for robust occlusion detection and occlusion map generation. The occlusion map thus obtained is used for developing a novel observation model update mechanism to avoid the model degradation. A patch occlusion ratio is employed in the calculation of the confidence score to improve the tracking performance. Quantitative and qualitative performance evaluations on the two above mentioned benchmark datasets demonstrate that this second proposed tracking algorithm generally performs better than several state-of-the-art methods and the first proposed tracking method do. Despite the improved performance of this second proposed tracking algorithm, there are still

some challenging attributes of OTB-50 and of VOT2016 for which the performance needs to be improved.

Finally, a third tracking algorithm is proposed by developing a scheme for collaboration between the discriminative and generative appearance models. The discriminative model is explored to estimate the position of the target and a new generative model is used to find the remaining affine parameters of the target. In the generative model, robust coding is extended to two dimensions and employed in the bilateral two dimensional PCA (2DPCA) reconstruction procedure to handle the non-Gaussian or non-Laplacian residuals by developing an IRRC technique. A 2D robust coding distance is introduced to differentiate the candidate sample from the one reconstructed from the subspace and used to compute the observation likelihood in the generative model. A method of generating a robust occlusion map from the weights obtained during the IRRC technique and a novel update mechanism of the observation model for both the kernelized correlation filters and the bilateral 2DPCA subspace are developed. Quantitative and qualitative performance evaluations on the two datasets demonstrate that this algorithm with HOG features generally outperforms the state-of-the-art methods and the other two proposed algorithms for most of the challenging attributes.

ಶ್ರೀ ವೀತರಾಗಾಯ ನಮಃ

ತಾಯಿ, ತಂದೆ
ಮತ್ತು
ಗುರುಗಳಿಗೆ ಸಮರ್ಪಣೆ

Acknowledgments

I would like to express my deep sense of gratitude and reverence to my supervisors, Dr. M.N.S. Swamy and Dr. M. Omair Ahmad, for their unflagging support, encouragement and invaluable guidance at the every phase of this research. Their patience, enthusiasm and cooperation have not just fueled this research, but also imbibed in me the need for perfection when performing research as well as when communicating the results.

I sincerely thank Dr. Wei-Ping Zhu, Dr. Hassan Rivaz and other members of my thesis examining committee for their feedback and support.

I am very grateful to my supervisors for the financial support that I received from their grants awarded by Natural Sciences and Engineering Research Council (NSERC) of Canada and the Regroupement Stratégique en Microsystèmes du Québec (ReSMiQ). I would like to acknowledge the financial support that I received during my doctoral studies from Concordia University through bursaries, awards, teaching assistantships and reimbursement of open access publication charges. I gratefully acknowledge Ministre de l'Éducation, de l'Enseignement Supérieur et de la Recherche (MEESR) du Québec for awarding the Doctoral Research Scholarship to support my research work.

My passion to pursue the doctoral degree was sprouted during my M.Tech dissertation work that was supervised by Dr. S.V. Narasimhan. I would like to offer my deep admiration and gratitude to him for motivating and influencing me to pursue the doctoral degree, and provided unlimited support and invaluable guidance when I was looking for opportunities to pursue the doctoral degree. Also, I am highly indebted to Dr. Girish Chandra and Dr. S. D. Sudarsan for their kind support and co-operation, which helped me to secure an admission for the doctoral program at Concordia University.

I would like to thank all my fellow colleagues at the Center for Signal Processing and Communications, Concordia University, specially, Praveen Korrai, Soni Pratti-

pati, Mohamed Naiel, Mahdi Parchami, Mufleh Al-Shatnawi, Hamidreza Sadreazami, Marzieh Amini, Omid Saatlou and Masoumeh Abkenar. Much respect and appreciation to my friends, Rajeev Yadav, Arun Venkatesan, Vijay Badagi, Sarath Somasekharan, Prashanth Venkataswamy, Durai Chelvan and Srinivasan Jayaraman, and their families for their kindness and invaluable support during my stay at Montreal.

Finally, I thank my family members and relatives for their love and support. My achievements up to this instant would not have been possible without the blessings of my parents (Smt. Jwalanamma and Shri Kantharajappa). I would like to specially thank them for their unconditional love, continuous support and constant encouragement to excel in my studies, career and life. Special thanks to my brother (Suchendra), his wife (Suma) and kids (Jinag and Sannidhi) for their every possible support and kindness. My deepest gratitude goes to my wife Pooja for her patience, abundant love, unlimited support and constant encouragement throughout my graduate life. Last, but not the least, I would like to express my deep love for my lovely son, Pratham, whose presence has given me the happiness, strength and peace of mind.

Contents

List of Figures

List of Tables

List of Symbols

a, b, c and d	Distances in pixels
A,D,Q,T	No overlapping 8×8 sub-blocks
$\mathbf{A}$, $\mathbf{A}_t^m$	Weight matrices obtained by the IRRC for $\mathbf{Y}$ and $\mathbf{Y}_t^m$, respectively
$\mathbf{b}_i$	Sparse code of the i-th local patch
B,C,E,H,J,O,R,S	Two overlapping 8×8 sub-blocks
$\mathbf{B}$	Sparse codes of the image observation $\mathbf{Y}$
$\hat{\mathbf{B}}$	Sparse codes of the tracked target candidate $\hat{\mathbf{Y}}$
c_1, c_2, c_3, c_4, c_5	Constants
C^m	Confidence score of the candidate $\mathbf{Y}_t^m$
$\hat{C}$	Confidence score of tracked target candidate $\hat{\mathbf{Y}}$
$d(.)$	Distance metric
$d_{2DRC}(.)$	2D robust coding distance metric
d_g	The dimension of the observation vector $\mathbf{y}$, $\overline{\mathbf{y}}$
$d_{GRC}(.)$	Global robust coding distance metric
$d_{GSLS}(.)$	Global standard least squares distance metric
$d_{GSLS\_RC}(.)$	Distance metric $d_{GSLS}(.)$ computed using optimum PCA basis coefficients $\mathbf{z}_{opt}$ obtained by the IRRC
d_h, d_w	The size of an image patch $\mathbf{X}$ in pixels
d_l, d_r	The size of an image $\mathbf{Y}$, $\overline{\mathbf{Y}}$, $\mathbf{Y}_t$ and $\mathbf{Y}_t^m$ in pixels
d_n	Number of elements in a non-overlapped local patch

d_o	Number of elements in a overlapped local patch
$\mathbf{D}$	Patch dictionary
$\mathbf{e}$	Error or residual vector
$\hat{\mathbf{e}}_t$	Error vector of the tracked target candidate at time t
$\{\mathbf{e}_{LP}^i\}_{i=1}^{N_n}$	Set of the local error or residual vectors
$\mathbf{E}$	Error or residual matrix
$\hat{\mathbf{E}}_t$	Error matrix of the tracked target candidate $\hat{\mathbf{Y}}$ at time t
$\mathbf{E}_h$	Holistic reconstruction error
$\mathbf{f}_p$	Pooled feature vector of the candidate
$\hat{\mathbf{f}}_p$	Pooled feature vector of $\hat{\mathbf{Y}}$
f_Θ	Probability density function
F,G,K,L	Four overlapping 8×8 sub-blocks
$\mathcal{F}$, $\mathcal{F}^{-1}$	Fourier transform and its inverse
$\overline{\mathbf{g}}$	Response map of correlation filter
$\overline{\mathbf{g}}_t^1$, $\overline{\mathbf{g}}_t^2$	Response maps of KCF-1 and KCF-2, respectively, at time t
$\widetilde{\mathbf{g}}_t$	Final response map at time t
$\hat{\mathbf{h}}_j$	Contribution of $j-$th template in $\mathbf{T}$
$\mathbf{H}$	Correlation filter
$\mathbf{I}$	Identity matrix
$\hat{j}$	Location of the template to be replaced in $\mathbf{T}$
k_g	Number of the global and local PCA basis vectors
k_l, k_r	Number of B2DPCA left- and right-projection basis vectors
l_r, l_c	The size of a non-overlapped local patch
L_Θ	The objective function of error vector $\mathbf{e}$, error matrix $\mathbf{E}$
m	Index to identify the samples/particles
M	Number of samples/particles of the state variable $\mathbf{s}_t$
n_t	Number of target templates
$\mathcal{N}$	Gaussian distribution

N_n	Number of non-overlapped local patches inside the target region
N_o	Number of overlapped local patches extracted within the target region
$\mathbf{O}$, $\mathbf{O}_1$, $\mathbf{O}_2$	Binary occlusion maps
$\mathbf{O}_t$	Occlusion map obtained for the tracked target candidate at time t
$\hat{\mathbf{O}}_t$	Updated occlusion map of $\mathbf{O}_t$
$\overline{\mathbf{O}}_t$	Resized occlusion map obtained from $\hat{\mathbf{O}}_t$
O_{thr}	Precomputed threshold to detect occlusion
$p(\mathbf{s}_t\|\mathbf{s}_{t-1})$	State transition/dynamic model
$p(\mathbf{Y}_t\|\mathbf{s}_t)$	An observation model representing the likelihood of the observation $\mathbf{Y}_t$ at state $\mathbf{s}_t$
$p(\mathbf{Y}_t^m\|\mathbf{s}_t^m)$	The likelihood of the observation $\mathbf{Y}_t^m$ at state $\mathbf{s}_t^m$
$p(\mathbf{s}_t\|\mathcal{Y}_t)$	Posterior probability
$\mathbf{p}$	Updated tracked target sample
$\hat{\mathbf{p}}$	Estimated image of $\mathbf{p}$ using PCA subspace
$\mathbf{P}_i$	$i-$th overlapped local patch inside the target region
q	Iteration index
$\mathbb{R}$	Real space
R_T, R_G	Bounding boxes of the tracking result, the ground truth
$\mathbf{s}_t$	Target state
$\mathbf{s}_t^m$	m-th sample of the state $\mathbf{s}_t$
s_t	Scale at time t
t	Time
$\mathbf{T}$	Set of target templates, $\{\mathbf{T}_i\}_{i=1}^{n_t}$
T_{GP}, T_{BP}	Time instants
$\mathbf{U}_{BP}$, $\mathbf{V}_{BP}$	B2DPCA left- and right-projection matrices

$\mathbf{U}_{GP}$, $\mathbf{U}_{LD}$	Matrix of PCA basis vectors
$\{\mathbf{U}_{LP}^{i}\}_{i=1}^{N_n}$	Set of matrix of local PCA basis vectors
$\mathbf{v}_i$	Normalized feature vector of the i-th local patch
$\mathbf{V}_f$	Square matrix containing all the normalized feature vectors $\mathbf{v}_i$ of the candidate
$\mathbf{W}$, $\mathbf{W}_t^m$	Global diagonal weight matrices of $\mathbf{y}$ and $\mathbf{y}_t^m$, respectively
$\hat{\mathbf{W}}$	Global diagonal weight matrix of tracked target candidate $\hat{\mathbf{Y}}$
$\{\mathbf{W}_{LP}^{i}\}_{i=1}^{N_n}$	Set of the local diagonal weight matrices
$\mathbf{W}_{map}$	Weight map
x_t, y_t	Horizontal and vertical spatial translations at time t
$\mathbf{X}$, $\mathbf{X}^D$	An image patch of size $d_h \times d_w$ cropped out from the frame
$\hat{\mathbf{X}}$	The target appearance model
$\mathbf{X}_t$	Target appearance at time t
$\tilde{\mathbf{X}}_t$	Modified target appearance at time t
$\hat{\mathbf{X}}_t^1$, $\hat{\mathbf{X}}_t^2$	Target appearance models of KCF-1 and KCF-2, respectively, at time t
$\mathbf{X}_{i,j}$	Circular shift of $\mathbf{X}$ by i and j pixels vertically and horizontally, respectively
$\mathbf{y}$, $\mathbf{y}_t$, $\mathbf{y}_t^m$	An observation vector obtained from image observation matrices $\mathbf{Y}$, $\mathbf{Y}_t$ and $\mathbf{Y}_t^m$, respectively, of size $d_l \times d_r$
$\overline{\mathbf{y}}$	Centered image observation vector of $\mathbf{y}$
$\overline{\mathbf{y}}_t^m$	Centered candidate vector of $\mathbf{y}_t^m$
$\{\mathbf{y}_{LP}^{i}\}_{i=1}^{N_n}$	Set of non-overlapped local patch vectors
$\{\overline{\mathbf{y}}_{LP}^{i}\}_{i=1}^{N_n}$	Set of centered non-overlapped local patch vectors, $\{\overline{\mathbf{y}}_{LP}^{i} = \mathbf{y}_{LP}^{i} - \boldsymbol{\mu}_{LP}^{i}\}_{i=1}^{N_n}$
$\mathbf{y}_{OL}^{i}$	Vector obtained from the 2DDCT of local patch $\mathbf{P}_i$
$\mathbf{Y}$	An image observation matrix of size $d_l \times d_r$
$\overline{\mathbf{Y}}$	Centered image observation matrix of size $d_l \times d_r$

$\hat{\mathbf{Y}}$	Tracked target candidate image
$\tilde{\mathbf{Y}}$	Matrix obtained from the vectors $\mathbf{y}_{OL}^{i}$
$\mathbf{Y}_r$	Reconstructed holistic image
$\mathbf{Y}_t$	An image matrix of size $d_l \times d_r$ observed at time t
$\mathbf{Y}_t^m$	Candidate image sample of size $d_l \times d_r$ observed by $\mathbf{s}_t^m$
$\mathcal{Y}$	A set of K image observations, $\{\mathbf{Y}_1, ..., \mathbf{Y}_K\}$
$\mathbf{z}$, $\mathbf{z}_t^m$, $\mathbf{q}$	Global PCA basis coefficients of $\mathbf{y}$, $\mathbf{y}_t^m$ and $\mathbf{p}$, respectively
$\mathbf{z}_{opt}$	Optimum global PCA basis coefficients
	Global PCA basis coefficients
$\{\mathbf{z}_{LP}^{i}\}_{i=1}^{N_n}$	Set of the local PCA basis coefficients
$\mathbf{Z}$	B2DPCA projection coefficients of size $k_l \times k_r$
$\mathbf{Z}_{opt}$, $\mathbf{Z}_{t,opt}^m$	Optimum B2DPCA projection coefficients of $\mathbf{Y}$ and $\mathbf{Y}_t^m$, respectively
α_t	Aspect ratio at time t
$\boldsymbol{\alpha}$	Correlation filter coefficients in kernel space
$\boldsymbol{\alpha}_t^1$, $\boldsymbol{\alpha}_t^2$	Learned coefficients of KCF-1 and KCF-2, respectively
$\hat{\boldsymbol{\alpha}}$	Learned coefficients model
$\hat{\boldsymbol{\alpha}}_t^1$, $\hat{\boldsymbol{\alpha}}_t^2$	Learned coefficients' models of KCF-1 and KCF-2, respectively, at time t
β_{GP}, δ_{GP}, β_{BP}, δ_{BP}	Parameters of the weight function
γ	Constant used in the observation likelihood $p(\mathbf{Y}_t^m \vert \mathbf{s}_t^m)$
λ_{GLP}	Penalty constant used in $d_{GRC}(.)$ and $d_{LRC}(.)$
λ_{BP}	Penalty constant used in $d_{2DRC}(.)$
$\boldsymbol{\mu}_{BP}$	B2DPCA mean matrix
$\boldsymbol{\mu}_{GP}$, $\boldsymbol{\mu}_{LD}$	PCA mean vectors
$\{\boldsymbol{\mu}_{LP}^{i}\}_{i=1}^{N_n}$	Set of local PCA mean vectors
ϕ_t	Skew direction at time t
Φ	Mapping function to a kernel space

ψ_{GP}, ψ_{BP}	Small positive scalars used to terminate IRRC
ρ_{Θ}	Negative logarithm of probability density function f_{Θ}
Σ	Diagonal covariance matrix
$\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2$	Variances of the affine parameters $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$
τ	Occlusion ratio
τ_1, τ_2, τ_{KCF}	Thresholds used in the observation model update
τ_{p_i}	Patch occlusion ratio
θ_t	Rotation angle at time t
Θ	The parameter that characterizes the distribution of error vector $\mathbf{e}$, error matrix $\mathbf{E}$
ξ	Correlation filter regularization parameter

List of Abbreviations

2DDCT	Two dimensional discrete cosine transform
2DPCA	Two dimensional principal component analysis
2DRC	Two dimensional robust coding
3DDCT	Three dimensional discrete cosine transform
ACLE	Average center location error
ACT	Adaptive color attributes
AOR	Average overlap rate
A-Rank	Accuracy rank
ASLA	Adaptive structural local sparse appearance
AUC	Area under curve
B2DPCA	Bilateral 2DPCA
B2DPCARC	Bilateral 2DPCA and robust coding
BC	Background clutter
CLE	Center location error
CN	Color names
CNN	Convolutional neural network
DCF	Discriminative correlation filters
DCT	Discrete cosine transform
DDL	Discriminative dictionary learning
Def	Deformation
DLR	Discriminative low-rank learning

DLT	Deep learning tracker
DSL	Discriminative subspace learning
EM	Expectation maximization
FM	Fast motion
FOB	Four overlapping blocks
GMM	Gaussian mixture model
GSLS	Global standard least squares
HOG	Histogram of oriented gradients
IL3DDCT	3DDCT based object representation and its incremental learning
IPR	In-plane rotation
IRRC	Iteratively reweighted robust coding
IV	Illumination variation
IVT	Incremental visual tracking
KCF	Kernelized correlation filters
L1APG	L1 tracker using accelerated proximal gradient algorithm
LASSO	Least absolute shrinkage and selection operator
LDSAM	Local 2DDCT sparse appearance model
LR	Low resolution
LSGPR	Locally structured Gaussian process regression
LSST	Least soft-threshold squares
LWIST	Locally weighted inverse sparse tracker
MAP	Maximum a posteriori
MB	Motion blur
MLE	Maximum likelihood estimation
NOB	No overlapping blocks
Occ	Occlusion
OPE	One-pass evaluation

OPR	Out-of-plane rotation
OR	Overlap rate
OTB-50	Object tracking benchmark-50
OV	Out-of-view
PCA	Principal component analysis
PCOM	Probability continuous outlier model
PDF	Probability density function
RC	Robust coding
R-Rank	Robustness rank
RSC	Robust sparse coding
SLS	Standard least squares
SPT	Sparse prototype tracker
SV	Scale variation
SVD	Singular value decomposition
SVM	Support vector machine
TOB	Two overlapping blocks
VOT2016	Visual object tracking 2016
WLCS	Weighted local cosine similarity
WLS	Weighted least squares
WRMPCA	Weighted residual minimization in PCA subspace

Chapter 1

Introduction

Visual tracking has seen a flurry of research in the last two decades due to its wide range of real-life applications including activity detection, action recognition, human behavior analysis, sports video analysis, vehicle navigation, human computer interaction, video indexing and retrieval, medical imaging, robotics, security and surveillance [4–6]. Visual tracking, the task of finding the location of an object in the subsequent frames of a video given the initial location, is a critical intermediate step for identifying and analyzing the anomalous activity or behavior in a security and surveillance videos, or for an autonomous operation of robots in the dynamic environments. In spite of much progress in the last three decades, visual tracking still remains a challenging problem due to the complexity involved in target searching as well as in handling intrinsic (e.g., pose changes, shape deformation) and extrinsic (e.g., varying viewpoints, rotation and scaling due to camera motion, illumination changes, occlusions, cluttered and moving backgrounds) object appearance variations [5–7]. These appearance variations should be handled carefully for a reliable tracking performance for which the appearance model should adapt to the intrinsic appearance variations and be robust enough for extrinsic appearance variations.

In this thesis, the focus is on single-camera, single-target, short-term, causal tracking. Single-camera, single-target tracking means there is only one object of interest being tracked in a sequence of image frames/video that is captured by a single cam-

era. The short-term tracking is defined by considering the length of the video that is used for tracking and the length of the video is in the order of a few seconds or a couple of minutes. In causal tracking, the target location at time t_k is estimated based on the information from earlier video frames at time t, where $t \leq t_k$.

1.1 Literature Review

In the literature, based on the representation scheme used to model the appearance of the object, the tracking algorithms are categorized into either generative or discriminative methods. Generative methods extract information only from the target region to model the object appearance and search for a region that is most similar to the target model. These methods are based on holistic templates [2, 8–14], local patches/fragments [11, 15–17], subspace models [1, 3, 18, 19] or local subspace models [20, 21]. Black *et al.* [22] proposed an optical flow framework for tracking, by learning a subspace model in offline mode to represent the target objects at predefined views and later on, extended their subspace representation to a mixture model to capture the object appearance effectively [23]. Jepson *et al.* [24] proposed an online appearance model using Gaussian mixture model (GMM) and expectation maximization (EM) algorithm for robust visual tracking. Kernel-based tracking using mean shift-based mode seeking procedure is proposed by Comaniciu *et al* [9]. Matthews *et al.* [10] proposed a template update method with Lucas-Kanade algorithm [8] to reduce the drifting problem by aligning with the first template. Wang *et al.* [25] proposed a GMM-based adaptive appearance model in a joint spatial-color space. Ross *et al.* [1] proposed an online learning of adaptive linear subspace to model the target appearance with a sample mean update and tracking the object in a particle filter framework. In visual tracking decomposition, the observation model is decomposed into multiple basic observation models to capture wide variations in illumination and pose [12]. As most of these methods use holistic model of target representation, they

cannot handle partial occlusion or background distracters.

On the other hand, the discriminative methods extract information not only from the target region, but also from the background to differentiate the target from the background within a local region by viewing the tracking as a binary classification problem (e.g., using boosting algorithms [26, 27], semi-supervised learning [28] and support vector machines (SVM) [29, 30]). Avidan [29] proposed an offline training of SVM classifier and extended it for object tracking in an optical flow framework. Collins *et al.* [31] proposed a variance ratio of the foreground and the background classes to find the discriminative features for object tracking. In ensemble tracking [32], a set of trained weak classifiers are combined to discriminate the target object from the background. Grabner *et al.* proposed an online boosting algorithm to update the discriminative features for tracking [26] and later on, proposed a semi-online boosting algorithm, which treats all the visual information from the tracking results as unlabeled data and adapts a classifier within the semi-supervised framework [28]. Further, a tracker based on multiple instance learning uses all the ambiguous positive and negative samples in the bags to learn a discriminative model to reduce the drifting problem [27]. The underlying structure of positive and negative samples are exploited in P-N algorithm to learn an effective classifier for the object tracking [33]. Wang *et al.* [34] proposed a superpixel tracker based on discriminative appearance model to distinguish between the target and the background. Further, an online discriminative feature selection algorithm is proposed which directly couples the classifier score with the sample importance, and optimizes the objective function in the steepest ascent direction for positive samples and steepest descent direction for negative samples [35]. Li *et al.* [36] proposed a compact Three Dimensional Discrete Cosine Transform (3DDCT)-based object representation and its incremental learning for robust visual tracking using a signal reconstruction-based similarity measure as the discriminative criterion to evaluate the likelihood.

Since the generative methods consider information from the target region alone

to model the appearance of the object, they are not efficient in cluttered environments, but they achieve a higher generalization performance with limited data [37]. In contrast to the generative methods, the discriminative methods perform better if the training set is large due to its capability of differentiating the target from the background [38]. The advantages of these individual methods are exploited by collaborating both generative and discriminative methods to model the appearance of the object in [39–46].

Trackers based on Sparse Representation: The success of sparse representation in vision applications, such as image denoising [47], image classification [48] and face recognition [49] has inspired Mei *et al.* to propose a l_1-tracker [2, 50], where a set of target templates and trivial templates are used to model the object appearance, and the target location is determined by solving an l_1-minimization problem. Further, the tracking robustness is improved by exploring the high-dimensional image features and group sparsity in [51]. In order to speed up the tracking process, the less expensive l_2-minimization is exploited to bound the l_1 approximation error without sacrifing the accuracy [52]. Further, L1 tracker using accelerated proximal gradient algorithm (L1APG) [13] was proposed to improve the l_1-tracker [2] in terms of both the speed and accuracy. Similar to l_1-tracker [2], the methods in [53, 54] use target template-based appearance model but with robust sparse coding (RSC) to account for non-Gaussian or non-Laplacian residuals by posing visual tracking as an iteratively reweighted residual minimization task. In [55], a two-stage online discriminative tracking algorithm with particle filter is proposed by representing the target object by local sparse codes and exploiting both the static and adaptive observation models.

Trackers based on Covariance Matrices: Although the above tracking methods based on sparse representation handle partial occlusion and pose changes relatively well, sometimes they fail to track the object with drastic appearance changes and background clutter. This is because most approaches depend on appearance at-

tributes of the object that are highly sensitive to the appearance variations, such as pose, shape and scale variations, of the object. To overcome these issues, the covariance region descriptor, which captures the spatial attributes from pixel coordinate values as well as the combination of different appearance attributes, such as color, image gradients, is explored in detection and classification [56], and in tracking [57–59]. Porikli *et al.* [57] employed an affine-invariant metric to find the best match by brute force search approach and to update the model by finding the intrinsic mean of the covariance matrices through Riemannian geometry. Further, Wu *et al.* proposed a probabilistic tracking on Riemannian manifolds [60] and later extended to fragments-based representation in [61]. Also, incremental covariance tensor learning is proposed to reduce the computational cost of the model update in [58]. Further, in Log-Euclidean Riemannian subspace learning algorithm [62], an eigenspace representation for each mode of the target is learned incrementally by updating the sample mean and eigenbasis adaptively.

Trackers based on Subspace Representation: Most of the trackers based on sparse representation employ the target templates to model the appearance of the object. In contrast, the subspace representation-based trackers exploit the rich and redundant image properties of the target templates to compactly represent the object appearance via basis vectors. In incremental visual tracking (IVT) [1], the object is represented in a low dimensional principal component analysis (PCA) subspace, which is learned and updated efficiently to adapt the appearance variations of the object. Further, incremental two dimensional PCA (2DPCA) has been used for object appearance model in visual tracking based on maximum likelihood estimation (MLE) [63]. Even though the schemes in [1] and [63] are effective in handling illumination and pose variations, they are sensitive to partial occlusion. This is due to the fact that the assumption of Gaussian distributed residual with small variances does not hold for object representation in visual tracking during partial occlusion. To address this issue, Wang *et al.* have exploited the strength of both the subspace and

sparse representations in [3] and [19] by introducing l_1-regularization into the PCA and 2DPCA reconstruction, respectively. Unlike [19], the method in [64] exploits the strength of both the subspace representation and RSC for visual tracking by introducing l_1-regularization into the 2DPCA reconstruction. The difference between [19] and [64] is in modeling the occlusion/outliers, the former method uses trivial templates and the latter uses weights to model the occlusion/outliers. In contrast to the IVT that assume Gaussian distributed residual with small variances, the method in [65] assume that the residuals are i.i.d. Gaussian-Laplacian distributed (i.e., an additive combination of i.i.d. Gaussian and i.i.d. Laplacian components) to handle outliers/occlusion effectively.

Trackers based on Discriminative Correlation Filters: Recently, discriminative correlation filters have gained significance in visual tracking due to its computational efficiency and tracking accuracy. In [66], adaptive correlation filters are learned to model the target appearance by minimizing the output sum of the squared error. Danelljan *et al.* have exploited adaptive color attributes in [67], and adaptive multi-scale correlation filters to handle scale variations of the object in [68, 69]. Henriques *et al.* have exploited the circulant structure of adjacent image patches in a kernel space based on intensity features [70], and histogram of oriented gradient (HOG) features [71] for visual tracking. Later, different variants of correlation filters have been proposed to increase the tracking performance using long-term memory components [72], target adaptation [73], complementary cues [74] and correlation particle filter [75]. In spite of the continuous improvement in tracking performance, standard discriminative correlation filter suffers from unwanted boundary effects due to the circulant assumption, thereby leading to a restricted target search region and a suboptimal training. The unwanted boundary problem has been addressed in [76–79] by learning a filter that has much smaller spatial support than the training samples. Learning a filter using a large spatial size training samples not only reduces the boundary effects [76], but also provides a large number of background training

samples [77]. However, these methods are computationally expensive due to their approach in solving the optimization problem.

Trackers based on Deep Features/Learning: In contrast to the discriminative correlation filter-based trackers [67, 68, 71, 80] that are learned using raw pixel values or handcraft features such as HOG [81], color attribute [82] or their combination, the trackers [78, 83–86] learned using deep convolutional neural network (CNN) features are more robust against geometric and photometric variations of the target. This is due to the high discriminative capability of deep CNN features that are obtained by training CNNs using a large scale ImageNet dataset [87, 88]. Even though the trackers [78, 83–86] that use deep CNN features achieve superior tracking performance, they do not exploit the full benefits of end-to-end learning of deep CNNs. In order to exploit the power of deep CNNs in visual tracking, a large-scale dataset specialized for visual tracking covering a wide range of variations in the target and background is needed to train CNNs. Due to lack of specialized data for tracking, most of the deep learning-based trackers [78, 83–86] have adapted a pre-trained CNN, which is trained for classification task, to obtain the deep CNN feature maps for visual tracking. But the deep feature representations thus obtained may not be effective due to the fundamental difference between classification and tracking problem. Moreover, the feature representations obtained by a pre-trained CNN may be less discriminative for tracking specific objects. In addition to this, the superior performance of deep learning-based trackers is achieved at the cost of excessive complexity and memory requirements. Considering these facts, in this thesis, non deep learning-based tracking algorithms are developed that do not rely on any external source of information or pre-trained model to obtain the features, but instead use the features obtained from the object itself to learn and update the object appearance model for tracking.

1.2 Motivation

In object tracking, the object representation schemes have received extensive attention from the research community along with the observation model update schemes in order to handle appearance variations of the object as well as occlusion. In general, the representation schemes are either generative or discriminative with each one having its own merits and demerits. As generative models exploit only the information from the target object and do not take the background into account, they are less effective in cluttered environments, but they achieve higher generalization with limited data. On the other hand, discriminative models distinguish the target region from the background and perform better provided the training set is large.

Most of the appearance models use raw pixels-based object representation except in some cases, where Haar features [26, 27], Gabor features [89–91], adopted features (e.g., intensity [1], color [92], texture [32]), description models (e.g., holistic histogram [9], part-based histogram [93]) are used. The exploration of other features/transform-domain features, which can reduce the dimensionality of the dictionary as well as the candidates in the object representation scheme have not been investigated. Further, during occlusion, the assumption that the residual/error follows either the Gaussian or the Laplacian distribution may not hold for object representation in visual tracking, as the residual cannot be modeled with small variances [3]. But in practice, the distribution of the residual may be different from the Gaussian or the Laplacian distribution especially during occlusions, corruptions and appearance variations of the object. In this regard, not much effort has been made to model the residual/error other than the Gaussian or the Laplacian distribution for the practical visual tracking scenarios. In the literature, only a few attempts have been made to combine both the generative and discriminative appearance models in a collaborative framework to harvest the benefits of these individual representation schemes [40, 41, 43–46]. These collaborative schemes have not yet matured enough to provide better tracking per-

formance, and hence, there is a need for a careful study on new ways of collaboration by considering the strengths of the generative and discriminative appearance models.

1.3 Objectives and Organization of the Thesis

The objective of the thesis is to develop visual object tracking algorithms by addressing the limitations of the existing methods to enhance the tracking performance by exploring different object representation schemes for the object appearance modeling and then devising mechanisms to update the observation models. In this thesis, three visual object tracking algorithms are proposed to improve the tracking performance over that of the existing methods.

The first tracking algorithm is based on the global appearance model using robust coding and its collaboration with a local model. The global PCA subspace is used to model the holistic appearance of the object, and then the optimum global PCA basis coefficients and the global weight matrix are computed by developing an iteratively reweighted robust coding (IRRC) technique. The global appearance model is collaborated with the local model to exploit the advantages of the individual models. Global and local robust coding distance measures are introduced to find the candidate sample having appearance similar to that of the reconstructed one from the subspace, and then these distances are used to define a novel observation likelihood. A robust occlusion map generation scheme and a mechanism to update both the global and local observation models are developed. Experiments conducted on the two popular benchmark datasets, the object tracking benchmark-50 (OTB-50) [7] and the visual object tracking 2016 (VOT2016) [94], show that this tracking algorithm with HOG features generally performs better than several state-of-the-art tracking methods do for most of the challenging attributes as per both quantitative and qualitative evaluations. However, there is a need to improve the tracking performance on some challenging attributes of OTB-50 and that of VOT2016.

Therefore, a second tracking algorithm based on a structural local two dimensional discrete cosine transform (2DDCT) sparse appearance model and an occlusion handling mechanism is proposed. The energy compaction property of 2DDCT is exploited in the object representation by using only a few 2DDCT coefficients in both the candidate samples and dictionary, which reduces the computational cost of l_1-minimization. The reconstruction of a holistic image from the overlapped local patches is presented, and the holistic image thus obtained is used along with the pooled feature vector to develop a robust occlusion map generation scheme. Using the occlusion map, a novel observation model update mechanism is developed to handle the appearance change of the object and to avoid the model degradation. A patch occlusion ratio is used in the confidence score computation to enhance the tracking performance. Experiments conducted on the two benchmark datasets bear out that the second tracking algorithm generally performs better than several state-of-the-art methods and the first tracking algorithm proposed in this thesis do, from both the quantitative and qualitative points of view. In spite of the improvement in the performance of the second tracking algorithm, still there are some challenging attributes of the two datasets for which performance is to be improved.

This issue is addressed by proposing a third tracking algorithm, where a collaborative scheme of the discriminative and generative appearance models is presented. In the discriminative model, two kernelized correlation filters are used to estimate the position of the target, and the remaining affine motion parameters of the target are found using a new generative model. The idea of using the discriminative and generative models to find the location and the remaining affine motion parameters of the target, respectively, is based on the intuitions that (1) the discriminative capability of a tracker plays an important role in estimating the location of the target rather than in finding the other affine motion parameters of the target and (2) the generative capability of a tracker plays a significant role in estimating the remaining affine motion parameters of the target. In the generative model, the robust coding

technique used in the first tracking algorithm is extended to two dimensions, and then employed in the bilateral 2DPCA reconstruction procedure to develop an IRRC technique. A 2D robust coding distance measure is defined to compute the similarity between the candidate and the reconstructed sample from the bilateral 2DPCA subspace, and then used in the observation likelihood computation. A robust occlusion map generation scheme and the observation model update mechanism of both the kernelized correlation filters and the bilateral 2DPCA subspace are developed. To evaluate the performance of the third tracking algorithm both quantitatively and qualitatively, experiments are conducted on the two datasets mentioned before and it is observed that the third algorithm outperforms the state-of-the-art methods and the first two proposed algorithms for most of the challenging attributes.

The thesis is organized as follows. Chapter 2 introduces the background material concerning particle filters, and the benchmark datasets and performance evaluation measures used for the performance comparison of the trackers.

Chapter 3 introduces a tracking algorithm based on the global appearance model using robust coding and its collaboration with a local model. A scheme to represent the object using the global and local PCA basis vectors, and robust coding is presented. Global and local robust coding distances are defined to find the similarity between the candidate sample and the reconstructed sample from the corresponding subspace. A method to generate robust occlusion map and a mechanism to update both the global and local observation models are developed. The performance of the first tracking algorithm with that of the state-of-the-art methods is compared by conducting experiments on two popular benchmark datasets, OTB-50 and VOT2016.

Chapter 4 presents the object tracking algorithm based on a structural local 2DDCT sparse appearance model and an occlusion handling mechanism. A structural local 2DDCT sparse appearance model, which exploits the energy compaction property of 2DDCT to represent the object by using only a few 2DDCT coefficients, is presented. A procedure to reconstruct the holistic image from the overlapped local

patches is introduced. Experiments are conducted on the two benchmark datasets to analyze the performance of the second tracking algorithm with that of the state-of-the-art methods and that of the first tracking algorithm.

Chapter 5 introduces a tracking algorithm based on the collaboration of the discriminative and generative appearance models. The discriminative appearance model is used to find the target location, and a new generative appearance model based on bilateral 2DPCA and 2D robust coding is used to estimate the remaining affine parameters of the target. A 2D robust coding distance metric is defined to find the similarity between the candidate and the reconstructed sample from the bilateral 2DPCA subspace. The observation model update mechanism of both the kernelized correlation filters and bilateral 2DPCA subspace is presented. Experiments are conducted on the same two datasets mentioned above for performance comparison of the third tracking algorithm with that of the state-of-the-art methods and that of the first two proposed tracking algorithms.

Finally, Chapter 6 concludes by highlighting the contributions of the thesis followed by suggestions for possible future work.

Chapter 2

Background Material

This chapter covers the background information that is helpful in understanding the proposed tracking algorithms and their performance evaluation in later chapters of this thesis.

Section 2.1 discusses the merits and demerits of principal component analysis (PCA), two dimensional PCA (2DPCA) and bilateral 2DPCA (B2DPCA) from the perspective of image representation. Section 2.2 briefly explains the particle filter framework for visual tracking. Section 2.3 presents the benchmark datasets and the evaluation measures used to compare the performance of the proposed methods with that of the other trackers.

2.1 Merits and Demerits of PCA, 2DPCA, and B2DPCA

PCA [95] is a well-established linear-dimension reduction technique and has been used extensively in the areas of pattern recognition [96], computer vision [97], signal processing [98] etc. PCA finds the projection directions by minimizing the reconstruction error in those directions and then projects the original data into a lower-dimensional space spanned by those directions corresponding to the top eigenvalues [99]. In image

representation, recognition and retrieval using PCA, a 2D image matrix is usually transformed into 1D long vector by concatenating either column by column or row by row that leads to a point in a high-dimensional vector space. In spite of many advantages due to the conversion from 2D to 1D, it suffers from the following problems. First, there is a requirement of a large number of training samples for a reliable and robust estimation about the characteristics of the data distribution as each image sample is represented as a point in a high-dimensional space. Second, PCA suffers from small sample size problem due to the limited availability of data in real-life applications like face recognition and image retrieval. Third, the exploitation of the spatial redundancies within the image ensembles is not possible due to absence of spatial information [99]. Because of large size and relatively small number of training samples, accurate computation of the covariance matrix is difficult [100], and then finding the eigenvectors of a large size covariance matrix is very time consuming.

These problems have been addressed in 2DPCA [100] by directly computing the eigenvectors from the image covariance matrix, which is computed from the image matrix itself without matrix-to-vector conversion. Since both the number of rows and columns of the image covariance matrix is equal to the width of the image, the size of the covariance matrix of 2DPCA is much smaller than that of PCA. Hence, 2DPCA has a more efficient computation of the eigenvectors with less computational time. As the unilateral projection (right-multiplication) scheme is adopted in 2DPCA, it is also termed as unilateral 2DPCA. When compared with PCA, 2DPCA requires a larger number of coefficients for image representation than that required by PCA [100]. Moreover, 2DPCA is also viewed as performing PCA on the row vectors of the image matrix, where the correlation among the column vectors of the image matrix has not been exploited. The weaknesses of 2DPCA have been overcome by B2DPCA [99, 101], where two sets of projection directions are simultaneously constructed to project the row and column vectors of the image matrices to two different subspaces, respectively.

For a $d_l \times d_r$ image $\overline{\mathbf{Y}}$ and $k_l \times k_r$ projection coefficients $\mathbf{Z}$, the bilateral projection is given by

$$\mathbf{Z} = \mathbf{U}_{BP}^{\top} \overline{\mathbf{Y}} \mathbf{V}_{BP} \tag{2.1}$$

and

$$\overline{\mathbf{Y}} = \mathbf{U}_{BP} \mathbf{Z} \mathbf{V}_{BP}^{\top} \tag{2.2}$$

where $\mathbf{U}_{BP} \in \mathbb{R}^{d_l \times k_l}$ and $\mathbf{V}_{BP} \in \mathbb{R}^{d_r \times k_r}$ represent orthogonal left- and right-projection matrices, respectively, and k_l and k_r are the number of B2DPCA left- and right-projection basis vectors, respectively. Given a set of K image observations $\mathcal{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_K\}$, the projection matrices $\mathbf{U}_{BP}$ and $\mathbf{V}_{BP}$ are computed as in [63, 99]. As the redundancies among the rows and the columns of the images are removed in B2DPCA, it requires less number of coefficients to represent an image than that required by 2DPCA [100].

2.2 Particle Filters

Generally, in the tracking methods based on particle filter framework, the target motion is estimated using a Markov model with hidden state variables [102]. In this thesis, the target motion between two consecutive frames is assumed to be affine. Let $\mathbf{s}_t$ denote a state variable describing the affine motion parameters of a target at time t. Given a set of image observations $\mathcal{Y}_t = \{\mathbf{Y}_1, ..., \mathbf{Y}_t\}$ at time t, the posterior probability is inferred recursively by the Bayes' theorem [1]

$$p(\mathbf{s}_t|\mathcal{Y}_t) \propto p(\mathbf{Y}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1})\, p(\mathbf{s}_{t-1}|\mathcal{Y}_{t-1})\, d\mathbf{s}_{t-1} \tag{2.3}$$

where $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ represents the dynamic (state transition) model, $p(\mathbf{Y}_t|\mathbf{s}_t)$ represents the observation model and $\mathbf{s}_t$ is the target state.

In this thesis, an affine transformation with six parameters is adopted to model the target state $\mathbf{s}_t = (x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t)$, where x_t, y_t, θ_t, s_t, α_t and ϕ_t denote horizon-

tal and vertical translations, rotation angle, scale, aspect ratio and skew direction at time t, respectively. The dynamic model describes the target motion between two consecutive frames and is modeled by the Gaussian distribution assuming the affine parameters to be independent, i.e., $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t; \mathbf{s}_{t-1}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ denotes a diagonal covariance matrix whose elements are the variances of the affine parameters $(\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2)$. These affine parameters are used to crop a sub-image from the current frame and then normalized to the size $d_l \times d_r$. The dynamic model randomly selects M samples (particles) of the state variable $\mathbf{s}_t$ given the state $\mathbf{s}_{t-1}$ at $t-1$, which are used to generate the target candidates $\mathbf{Y}_t^m$, where $m = 1, 2, ..., M$. The optimal state of the tracked target $\hat{\mathbf{s}}_t$ is determined by the following maximum a posteriori (MAP) estimation

$$\hat{\mathbf{s}}_t = \arg\max_{\mathbf{s}_t^m} p(\mathbf{Y}_t^m|\mathbf{s}_t^m) p(\mathbf{s}_t^m|\mathbf{s}_{t-1}),\ m = 1, 2, ..., M \tag{2.4}$$

where $\mathbf{s}_t^m$ denotes the m-th sample of the state $\mathbf{s}_t$, and $\mathbf{Y}_t^m$ represents the image sample observed by $\mathbf{s}_t^m$. The observation model represents the likelihood of the observation $\mathbf{Y}_t^m$ at state $\mathbf{s}_t^m$. This will be defined in the subsequent chapters that use particle filter framework for tracking.

2.3 Benchmark Datasets and the Performance Evaluation Measures

For an exhaustive performance evaluation of the state-of-the-art tracking algorithms, to identify their strengths and weaknesses, and to draw future research directions for the development of robust tracking algorithms, the tracking experiments have to be conducted on the benchmark datasets and then evaluated using standard performance measures. Also, the benchmark datasets should contain all the individual challenging tracking scenarios and their combination including but not limited to illumination

changes, varying viewpoints, rotation and scaling due to camera motion, occlusions, cluttered and moving backgrounds etc. From this point of view, the experiments are conducted on two popular benchmark datasets, object tracking benchmark-50 (OTB-50) [7] and visual object tracking 2016 (VOT2016) [94], to compare the performance of the proposed trackers with that of other methods. The sequences from these benchmarks cover most of the real-life challenging situations in object tracking, such as motion blur due to fast movement, pose variation, complex background, varying lighting conditions, low contrast, scale change, heavy occlusion, in-plane and out-of-plane rotation.

The OTB-50 benchmark [7] consists of 50 challenging sequences (with 51 target objects) that are fully annotated with bounding boxes as well as 11 different attributes, namely, illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (Occ), deformation (Def), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background clutter (BC) and low resolution (LR). In OTB-50, there are 18 and 7 sequences with more than 500 and 1000 frames, respectively. The VOT2016 benchmark [94] consists of 60 challenging sequences that are fully annotated with rotated bounding boxes as per the target rotations. This is in contrast to the upright bounding boxes annotated in OTB-50. Further, VOT2016 dataset sequences are per-frame annotated with 5 visual attributes that reflect a particular challenge in appearance attributes, such as camera motion, illumination change, motion change, occlusion and size change. If a particular frame does not contain any of these five attributes, then it is denoted as empty. Again, this is also in contrast to OTB-50, where the sequences are annotated globally with attributes even though they may occupy only a few frames in a sequence. Recently, it has been shown that the performance measures computed from the global attribute annotations, like in OTB-50, are significantly biased toward the dominant attributes in the sequences, whereas the bias is significantly reduced with per-frame annotation, like in VOT2016, even in the presence of mis-annotations [103]. In VOT2016, there

are 14 and 2 sequences with more than 500 and 1000 frames, respectively.

In general, two frame-based metrics, namely, overlap rate (OR) and center location error (CLE), are employed to evaluate the tracker in a given frame. The OR is defined using a score in the Pascal visual object classes challenge [104] as $\text{OR} = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$, where R_T and R_G are the bounding boxes of the tracking result and the ground truth, respectively, and $area(R)$ denotes the area of the region R. In case of OR, its value should be as high as possible, approaching one for better tracking performance. CLE is the relative distance (in pixels) between the center positions of the tracking result and the ground truth. For better tracking performance, the value of CLE should be as close to zero as possible. Further, average OR (AOR) and average CLE (ACLE) are used to measure the overall performance of the tracker for a given sequence. However, when the tracker loses the target, the output location and size of the bounding box can be random, and hence, ACLE and AOR may not evaluate the tracking performance efficiently [105]. The authors of [7] and [94] have used other performance measures [105, 106] derived from the two basic metrics, OR and CLE, to analyze the performance of some existing trackers on their benchmark datasets, OTB-50 and VOT2016, respectively. Likewise, in this thesis, the performance measures used in [7] and [94] are employed to evaluate and compare the proposed trackers with the state-of-the-art methods on the OTB-50 and VOT2016 benchmark datasets, respectively.

In OTB-50, the performance of a tracker for a given sequence is evaluated using the *success rate* and the *precision score* [105]. The former is the ratio of successful frames whose OR is larger than a given threshold, to the total number of frames in a sequence. On the other hand, the later is the percentage of frames whose CLE is less than a given threshold distance of the ground truth. By using multiple thresholds, two curves are obtained showing how the threshold value affects the *success rate* and the *precision score*, and are called as *success plot* and *precision plot*, respectively, for a given sequence. Further, these *success curves* and *precision curves* are averaged over

all the sequences to obtain the overall *success plot* and *precision plot*, respectively. In order to quantify the overall performance of a tracker, the area under curve (AUC) of the *success plot* or the *precision score* for the threshold of 20 pixels is employed. In OTB-50, the trackers are evaluated on the test sequences with an initialization from the ground truth position in the first frame and the overall *precision plot* or *success plot* at the end is reported. This evaluation is referred as one-pass evaluation (OPE).

In VOT2016, the performance of a tracker is analyzed using the accuracy and robustness. The accuracy is the average overlap between the predicted and ground truth bounding boxes during successful tracking periods, whereas the robustness measures the number of times the tracker fails to track. In VOT2016, whenever a tracker predicts a bounding box with zero overlap with the ground truth, a failure is detected and the tracker is re-initialized. All the trackers are evaluated 15 times on each sequence and then per-frame accuracy is obtained as an average over these runs. Averaging per-frame accuracies gives per-sequence accuracy. On the other hand, per-sequence robustness is computed by averaging failure rates over different runs [94]. Further, the tracking results are ranked according to the accuracy and robustness performance metrics, and are called accuracy rank and robustness rank, respectively. Note that as the trackers with statistically equivalent results are merged while ranking, the different trackers may have the same accuracy rank and robustness rank [94]. In order to quantify the overall performance of a tracker, different averaging methodologies are used in VOT2016. The averages of the per-attribute results in an equal or weighted manner are denoted as mean and weighted mean, and the per-frame averaging of the results of the super-sequence is obtained by concatenating all of the sequences as pooled.

2.4 Summary

In this chapter, the advantages and disadvantages of PCA, 2DPCA and B2DPCA from the image representation point of view has been explained. Also, the particle filter framework for visual tracking has been introduced. Finally, the benchmark datasets used for the experimentation and the evaluation measures used to compare the performance of the proposed trackers have been presented. A novel tracking algorithm based on the global appearance model using robust coding and its collaboration with a local model will be presented in the next chapter.

Chapter 3

The Global Appearance Model using Robust Coding and its Collaboration with a Local Model for Visual Tracking

3.1 Introduction

In incremental visual tracking (IVT) [1], the object is represented by a low dimensional principal component analysis (PCA) subspace (Figure 3.1a), which is learned and updated efficiently to adapt to the object appearance variations. Even though the PCA subspace representation is effective in handling illumination and pose variations, it is sensitive to partial occlusion. This is because the assumption of Gaussian distributed residual with small variances (i.e., small dense noise) does not hold for representation of the object in visual tracking during partial occlusion. Further, the IVT does not have a mechanism to detect and remove the occlusion/outliers while updating its observation model with new observations. Similar drawbacks are observed in other tracking algorithms based on Gaussian noise assumption or ordinary least squares method [62, 107]. The success of sparse representation has motivated Mei *et al.* [2] to propose a novel l_1-tracker using a set of target and trivial templates to model, respectively, the object appearance and to handle occlusions (Figure 3.1b).

Wang *et al.* [3] exploit the strength of both the subspace and sparse representations in sparse prototype tracker (SPT) by introducing l_1-regularization into the PCA reconstruction. Here, the target appearance and occlusion are modeled by PCA basis vectors and trivial templates, respectively, as shown in Figure 3.1c, and given by

$$\mathbf{y} = \boldsymbol{\mu}_{GP} + \mathbf{U}_{GP}\,\mathbf{z} + \mathbf{e} = \boldsymbol{\mu}_{GP} + [\ \mathbf{U}_{GP}\ \mathbf{I}\ ] \begin{bmatrix} \mathbf{z} \\ \mathbf{e} \end{bmatrix} \tag{3.1}$$

where $\mathbf{y} \in \mathbb{R}^{d_g \times 1}$ is an observation vector obtained from the observed image sample $\mathbf{Y} \in \mathbb{R}^{d_l \times d_r}$, $\mathbf{U}_{GP} \in \mathbb{R}^{d_g \times k_g}$ denotes a matrix of PCA basis vectors, $\boldsymbol{\mu}_{GP} \in \mathbb{R}^{d_g \times 1}$ represents mean vector, $\mathbf{z} \in \mathbb{R}^{k_g \times 1}$ represents the PCA basis coefficients, $\mathbf{e} \in \mathbb{R}^{d_g \times 1}$denotes the error or residual vector, and k_g and d_g are the number of PCA basis vectors and the dimension of the observation vector, respectively.

The holistic/global appearance model provides a compact representation of the target object and deals with global appearance variation of the object, but it cannot handle occlusion, as the partial and spatial information is not exploited to the full extent, thus leading to a poor performance of the tracker [1]. On the other hand, the local appearance models [20, 21, 108] exploit the partial and spatial structural information of the target, and hence, perform well during occlusions and background clutter. However, they do not provide a compact representation of the target like the holistic model does, and fail during global appearance change.

In this chapter, a new tracking algorithm based on the global appearance model using robust coding (RC) and its collaboration with a local model is proposed [109]. To model the global appearance of the object, the global PCA subspace is used and an iteratively reweighted robust coding (IRRC) technique is developed to compute the optimum global PCA basis coefficients and the global weight matrix [110]. A collaborative scheme of the global (holistic) and local (patch/block) appearance models is presented to capture their individual advantages. Global and local RC distances are defined to find the similarity between the candidate sample and its reconstructed

sample from the subspace. These RC distances are used to define the observation likelihood, which is in turn used to find the tracking result by the maximum a posteriori (MAP) estimation. As the weights obtained during IRRC capture the outlier/occlusion information, they are exploited to detect outliers/occlusions as well as to generate the occlusion map. The occlusion map thus generated is used to obtain occlusion-free samples that are accumulated for observation model update. Experiments conducted on the two popular benchmark datasets with comparison to the state-of-the-art tracking methods bear out the effectiveness and competency of the proposed algorithm for visual tracking.

This chapter is structured as follows. Section 3.2 introduces the representation of the object via global and local PCA basis vectors, and robust coding. Section 3.3 presents the proposed tracking algorithm. Experimental results are given and discussed in Section 3.4 followed by in Section 3.5 a summary of the work presented in this chapter.

3.2 Object Representation via Global and Local PCA Basis Vectors, and Robust Coding

In IVT [1], the residual is assumed to be Gaussian distributed with small variances, but this assumption may not holds good during partial occlusions as the residual cannot be modeled with small variances. On the other hand, if the residual is assumed to be Laplacian distributed, then it aims to handle outliers. But, in real practice these assumptions may not hold well especially in tracking during occlusion and corruption.

The advantage of robust sparse coding (RSC) is exploited in [53, 54, 64] for visual tracking to account for non-Gaussian or non-Laplacian residuals by posing visual tracking as an iteratively re-weighted residual minimization task. Similar to l_1-tracker [2], the trackers in [53, 54] use target template-based appearance model. As the target templates are coherent, the coding coefficients are sparse and hence, require l_1-norm

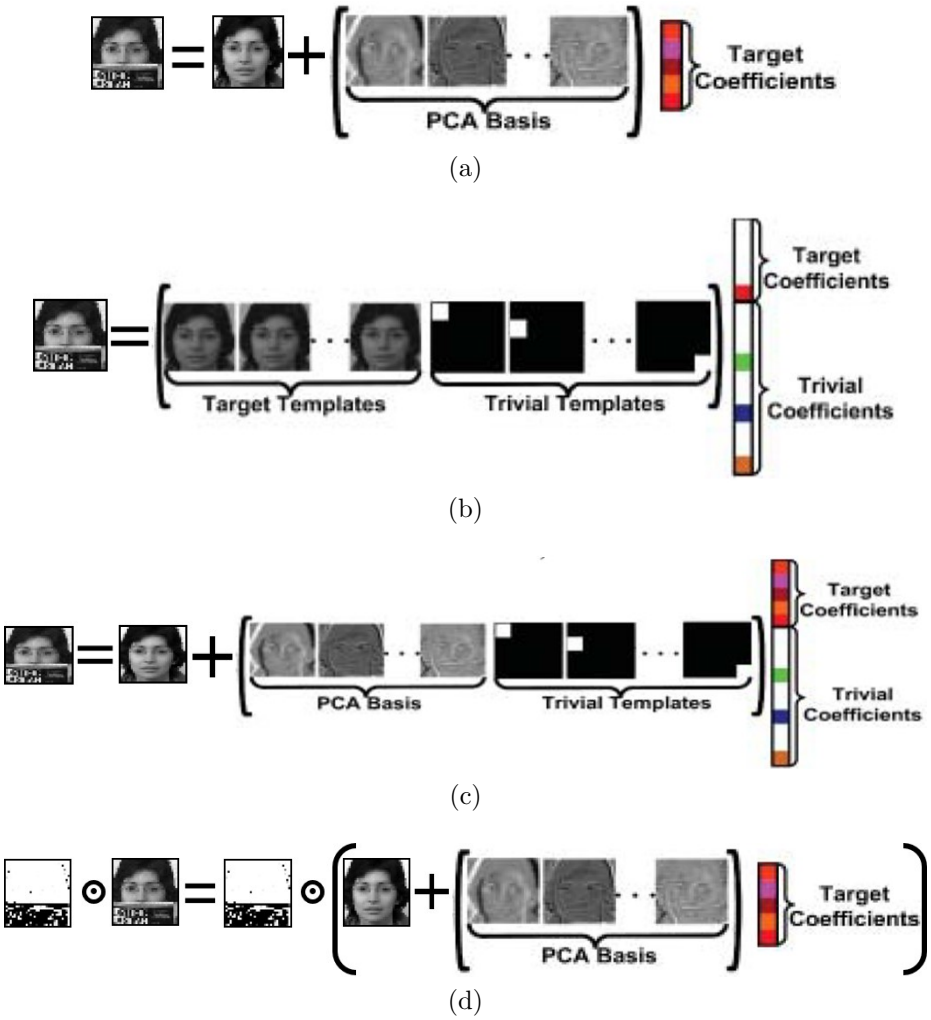


Figure 3.1: Different appearance models of (a) IVT [1], (b) l_1-tracker [2], (c) SPT [3], and (d) the proposed tracking algorithm, where $\odot$ indicates Hadamard product (element-wise product).

constraint on the coding coefficients. Since these trackers do not use the trivial templates to handle the occlusion, the dictionary size is greatly reduced. This reduction in dictionary size should reflect in the proportionate reduction of computational time, which does not happen, since each iteration of the residual minimization has to solve for a weighted least absolute shrinkage and selection operator (LASSO) problem and hence, not viable for time critical applications. The major differences between the trackers in [53, 54] are the template update mechanism and the weight function used

in RSC. Further, the strengths of both the subspace representation and RSC were exploited in [64] for visual tracking by introducing l_1-regularization into the bilateral two dimensional PCA (B2DPCA) reconstruction. In [64], the object appearance model is represented by the B2DPCA projection matrices and its observation model update mechanism is similar to that in [54]. The drawback of [64] is that the unnecessary imposition of l_1-norm constraint on the projection coefficients even though they are not sparse due to the orthogonality of the B2DPCA projection matrices. Unlike the appearance models of [53, 54, 64], which use RSC, the proposed appearance model uses RC as the PCA basis coefficients are not sparse. Also, the appearance model used in [64] is based on B2DPCA in contrast to that of classical PCA used in this chapter. Further, [54, 64] exploit only holistic information and differ in the update of the observation model. Also, the simplified version of the weight function is used in [54, 64] as compared to that in the proposed appearance model. Further, the algorithm in [111] uses weights for the different trackers, each with different features in a PCA subspace, and the weight for each tracker is computed using the tracking error between the forward-backward tracking. In contrast to the proposed algorithm, the algorithm in [111] does not have any mechanism to handle occlusions/outliers both in observation likelihood and observation model update. Further, in [112], the temporal appearance model using PCA basis vectors and the spatial constraint model using K-nearest candidate samples are collaborated for visual tracking. The main differences between the proposed algorithm and the algorithm in [112] are as follows. (1) The appearance model in the proposed algorithm is a generative one and [112] uses both the generative and discriminative models in collaboration. (2) There are variations in the observation likelihood computation and the observation model update. In [113], tracking by the correlation filter and online learning is combined in a way such that the tracking by online learning based on logistic regression classifier is used as a re-detection module and is activated when the correlation score of the correlation filter is less than a threshold which may otherwise result in tracking drift

or failure. As opposed to the proposed generative appearance model, the algorithm in [113] uses discriminative appearance models in both tracking by correlation filter and online learning. Also, the appearance model of the correlation filter in [113] does not take care of occlusions/outliers during its model update, which may degrade the appearance model, thereby resulting in frequent tracking drifts. In [114], the low-rank sub-dictionary to model the target and background appearance are learned independently, and these sub-dictionaries are used to represent the candidate samples. The observation likelihood is computed based on the reconstruction error of the candidate samples using these sub-dictionaries with their respective representation coefficients. The appearance model update of the proposed algorithm is different from that in [114] due to its discriminative appearance model.

3.2.1 Global Appearance Model using Robust Coding

Considering the visual tracking from the Bayesian estimation point of view, the PCA basis coefficients $\mathbf{z}$ can be computed by maximizing the posterior probability $p(\mathbf{z}|\mathbf{y})$. Assuming a uniform prior, the basis coefficients $\mathbf{z}$ are obtained by $\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} p(\mathbf{y}|\mathbf{z}) = \arg\max_{\mathbf{z}} p(\overline{\mathbf{y}} - \mathbf{U}_{GP}\,\mathbf{z}) = \arg\max_{\mathbf{z}} p(\mathbf{e})$, which is the maximum likelihood estimation (MLE), where $\overline{\mathbf{y}} = \mathbf{y} - \boldsymbol{\mu}_{GP}$ is the centered image observation vector and $\mathbf{e} = \overline{\mathbf{y}} - \mathbf{U}_{GP}\,\mathbf{z}$ is the error or residual vector. To find the MLE solution of the basis coefficients, the PCA basis matrix $\mathbf{U}_{GP}$ is written as $\mathbf{U}_{GP} = [\mathbf{u}_1; \mathbf{u}_2; ...; \mathbf{u}_{d_g}]$, where the row vector $\mathbf{u}_j$ is the j-th row of $\mathbf{U}_{GP}$, and the coding residual as

$$\mathbf{e} = \overline{\mathbf{y}} - \mathbf{U}_{GP}\,\mathbf{z} = [e_1; e_2; ...; e_{d_g}] \tag{3.2}$$

where each element of the residual is written as $e_j = \overline{y}_j - \mathbf{u}_j\mathbf{z}, j = 1, 2, ..., d_g$. Assuming that the elements of the residual $e_1, e_2, ..., e_{d_g}$ are independently and identically distributed (i.i.d) according to some probability density function (PDF) $f_{\Theta}(e_j)$, where $\boldsymbol{\Theta}$ denotes the unknown parameter that characterizes the distribution, the

likelihood of the estimator is given by $p(\mathbf{e}) = \prod_{j=1}^{d_g} f_\Theta(e_j)$. Maximization of this likelihood is equivalent to minimizing $L_\Theta(e_1, e_2, ..., e_{d_g}) = -\ln p(\mathbf{e}) = \sum_{j=1}^{d_g} \rho_\Theta(e_j)$, where $\rho_\Theta(e_j) = -\ln f_\Theta(e_j)$. Now, the MLE of basis coefficients $\mathbf{z}$, termed as robust coding (RC), can be written as

$$\min_{\mathbf{z}} \sum_{j=1}^{d_g} \rho_\Theta(e_j) = \min_{\mathbf{z}} \sum_{j=1}^{d_g} \rho_\Theta(\overline{y}_j - \mathbf{u}_j \mathbf{z}) \,. \tag{3.3}$$

Here, RC is introduced to account for non-Gaussian or non-Laplacian residuals/noises and reduce their effect on the computation of the PCA basis coefficients.

If the distribution f_Θ (or ρ_Θ) is known, MLE of the basis coefficients $\mathbf{z}$ can be computed by solving (3.3). Now the question is how to determine the unknown PDF f_Θ (or ρ_Θ). Explicitly assuming f_Θ as Gaussian or Laplacian distribution is simple, but not effective enough in visual tracking during corruptions/occlusions. Instead of determining ρ_Θ directly to solve (3.3), the assumptions of ρ_Θ mentioned in [115] are used to approximate the minimization problem in (3.3) by a weighted least squares (WLS) problem, given by

$$\min_{\mathbf{z}} \frac{1}{2} \| \mathbf{W}^{\frac{1}{2}} (\overline{\mathbf{y}} - \mathbf{U}_{GP} \, \mathbf{z}) \|_2^2 \,, \tag{3.4}$$

where $\mathbf{W}$ is a diagonal weight matrix representing the different noise types, and its element $\mathbf{W}_{j,j}$ is the weight assigned to each element of observation vector obtained from the observed image sample $\mathbf{Y}$ depending on the value of the residual. Since the weight matrix $\mathbf{W}$ is unknown and needs to be estimated, WLS in (3.4) is a local approximation of RC in (3.3). Therefore, the minimization procedure of RC can be converted into an IRRC problem with $\mathbf{W}$ being updated using the residuals in the previous iteration. Due to the properties of $\rho_\Theta(e_j)$ and its relation with $\mathbf{W}_{j,j}$, each $\mathbf{W}_{j,j}$ will be a non-negative scalar [115]. So the WLS problem in each iteration of IRRC is a convex problem and its solution can be found easily. Since the distribution

ρ_Θ is unknown, it is difficult to find the weight matrix $\mathbf{W}$. Thus, a logistic function given by

$$\mathbf{W}_{j,j} = \frac{\exp\left(\delta_{GP}\left[\beta_{GP} - e_j^2\right]\right)}{1 + \exp\left(\delta_{GP}\left[\beta_{GP} - e_j^2\right]\right)}, \tag{3.5}$$

where δ_{GP} controls the decreasing rate from 1 to 0, and β_{GP} controls the location of the demarcation point, is chosen as the weight function, as it satisfies the following properties: (1) weight assigned to each pixel of the observed image vector $\mathbf{y}$ depends on the corresponding value of the residual $\mathbf{e}$ and (2) the weight function has higher capability to classify inliers and outliers [115]. This weight function is bounded in [0,1] and adaptively assigns low weights to the outliers (usually the pixels with big residuals) to reduce their effects on the regression estimation which in turn reduces the sensitivity to outliers.

3.2.2 Iteratively Reweighted Robust Coding

The minimization problem in (3.4) can be solved by estimating iteratively the diagonal weight matrix $\mathbf{W}$ using (3.5) and the PCA basis coefficients $\mathbf{z}_{opt}$ using the following equation

$$\mathbf{z}_{opt} = \left(\mathbf{U}_{GP}^{\mathsf{T}}\,\mathbf{W}\,\mathbf{U}_{GP}\right)^{-1}\mathbf{U}_{GP}^{\mathsf{T}}\,\mathbf{W}\,\overline{\mathbf{y}}\,. \tag{3.6}$$

The estimation of $\mathbf{W}$ and $\mathbf{z}_{opt}$ recursively is termed as iteratively reweighted robust coding (IRRC) technique and is given in Procedure 3.1.

For practical applications, the diagonal weight matrix $\mathbf{W}$ is usually not known. Hence, the diagonal weight matrix $\mathbf{W}$ is set to Identity matrix $\mathbf{I}$ initially assuming that all the pixels are inliers (equivalently the residual $\mathbf{e} = \mathbf{0}$). Due to this, (3.6) reduces to $\mathbf{z} = \mathbf{U}_{GP}^{\mathsf{T}}\,\overline{\mathbf{y}}$ as $\mathbf{U}_{GP}^{\mathsf{T}}\,\mathbf{U}_{GP} = \mathbf{I}$. Now, the residual error $\mathbf{e}$ is obtained using (3.2). Then, the diagonal weight matrix $\mathbf{W}$ is computed using (3.5), and is used to reduce the effects of outliers on the estimation of PCA basis coefficients $\mathbf{z}$ using (3.6) so that the sensitivity to outliers can be greatly reduced. The above procedure is

repeated using (3.2), (3.5) and (3.6) until convergence or termination.

Procedure 3.1 Computation of $\mathbf{z}_{opt}$ and $\mathbf{W}$ by the iteratively reweighted robust coding technique

---

Input: Centered image observation vector $\overline{\mathbf{y}}$, PCA basis vectors $\mathbf{U}_{GP}$, and parameters of weight function δ_{GP} and β_{GP}.

1: Initialize $q = 0$ and $\mathbf{W}^q = \mathbf{I}$

2: Compute basis coefficients $\mathbf{z}^q = \mathbf{U}_{GP}^{\mathsf{T}}\,\overline{\mathbf{y}}$

3: Iterate

4: $\quad q \leftarrow q + 1$

5: $\quad$ Compute residual $\mathbf{e}^q = \overline{\mathbf{y}} - \mathbf{U}_{GP}\,\mathbf{z}^{q-1} = [e_1^q; e_2^q; ...; e_{d_g}^q]$

6: $\quad$ Compute the weights using

$$\mathbf{W}_{j,j}^q = \frac{\exp\left(\delta_{GP}[\beta_{GP} - (e_j^q)^2]\right)}{1 + \exp\left(\delta_{GP}[\beta_{GP} - (e_j^q)^2]\right)};\ j = 1, 2, .., d_g$$

7: $\quad$ Recompute $\mathbf{z}^q = \left(\mathbf{U}_{GP}^{\mathsf{T}}\,\mathbf{W}^q\,\mathbf{U}_{GP}\right)^{-1}\mathbf{U}_{GP}^{\mathsf{T}}\,\mathbf{W}^q\,\overline{\mathbf{y}}$

8: Until convergence or termination

Output: Basis coefficients $\mathbf{z}_{opt}$, diagonal weight matrix $\mathbf{W}$.

---

a) Convergence of the IRRC

Weighted least squares in (3.4) is a local approximation of RC in (3.3), and the objective function of (3.3) will be reducing for each iteration by the IRRC technique. The iterative minimization procedure in IRRC will converge to a global minimum solution as the original cost function of (3.3) is lower bounded (≥ 0) and is achieved when the following criterion is satisfied:

$$\frac{\|\mathbf{W}^q - \mathbf{W}^{q-1}\|_2}{\|\mathbf{W}^{q-1}\|_2} < \psi_{GP}\,, \tag{3.7}$$

where ψ_{GP} is a small positive scalar and q is the iteration index. Figure 3.2 shows the weights obtained for inlier and outlier pixels during the iteration and the convergence of (3.7) for one of the object tracking benchmark-50 (OTB-50) benchmark sequences (*Faceocc1*, frame #52).

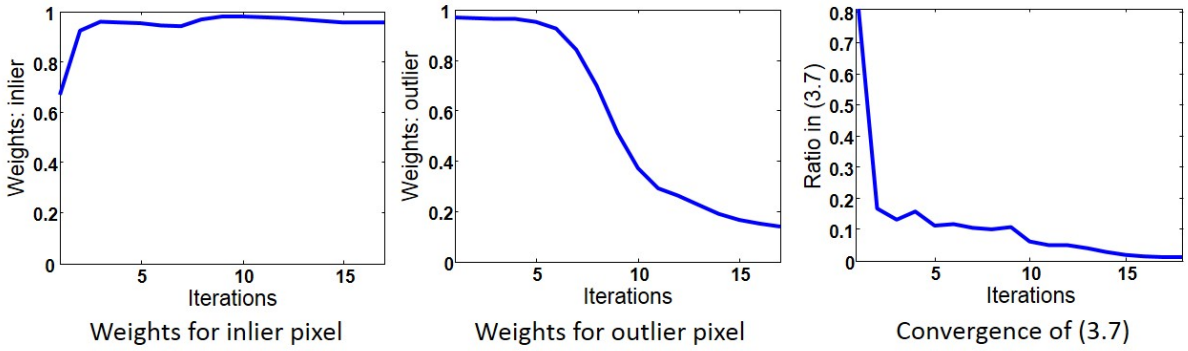


Figure 3.2: Weights for inlier and outlier pixels during the iteration and convergence of (3.7) for one of the OTB-50 benchmark sequences (*Faceocc1*, frame #52).

3.2.3 Global and Local PCA Subspace Models

In the proposed algorithm [109], the advantages of both the global and local representations are exploited by collaborating the global and local PCA subspace models in a generative tracking framework. This is unlike other collaborative methods [43, 46], which use local and global representations in generative and discriminative frameworks, respectively. Also, the strengths of both the subspace learning and RC are exploited by using the global PCA basis vectors $\mathbf{U}_{GP}$ and a global weight matrix $\mathbf{W}$ to model the object appearance and occlusion respectively, as shown in Figure 3.1d and is given by

$$\mathbf{W}^{\frac{1}{2}}\mathbf{y} = \mathbf{W}^{\frac{1}{2}}\left(\boldsymbol{\mu}_{GP} + \mathbf{U}_{GP}\,\mathbf{z}\right). \tag{3.8}$$

The reason for using the diagonal weight matrix $\mathbf{W}$ to account for occlusion/outliers rather than trivial templates as in [2, 3, 13, 65] is justified, since the weight function (3.5) assigns larger weights to inliers and smaller weights to outliers/occluded pixels as shown in Figure 3.2.

Similar to the global appearance model in PCA subspace, the local appearance

model in local PCA subspace is given by

$$\left(\mathbf{W}_{LP}^{i}\right)^{\frac{1}{2}}\ \mathbf{y}_{LP}^{i} = \left(\mathbf{W}_{LP}^{i}\right)^{\frac{1}{2}}\left(\boldsymbol{\mu}_{LP}^{i} + \mathbf{U}_{LP}^{i}\,\mathbf{z}_{LP}^{i}\right);\ i = 1, 2, ..., N_n \tag{3.9}$$

where $\mathbf{y}_{LP}^{i} \in \mathbb{R}^{d_n \times 1}$ denotes a i-th local patch image observation vector of the candidate sample, $\mathbf{U}_{LP}^{i} \in \mathbb{R}^{d_n \times k_g}$ represents a matrix of local PCA column basis vectors corresponding to the i-th local patch, $\boldsymbol{\mu}_{LP}^{i} \in \mathbb{R}^{d_n \times 1}$ denotes the mean vector corresponding to the i-th local patch, $\mathbf{z}_{LP}^{i} \in \mathbb{R}^{k_g \times 1}$ represents the local PCA basis coefficients, $\mathbf{W}_{LP}^{i} \in \mathbb{R}^{d_n \times d_n}$ is the local diagonal weight matrix corresponding to the i-th local patch, d_n is the number of elements in the local patch, k_g the number of local PCA basis vectors, and N_n is the number of non-overlapped local patches inside the target region. These non-overlapped local patches of size $l_r \times l_c$ pixels are extracted with a spatial layout as shown in Figure 3.3 and then vectorized to get the set of local patch observation vectors $\mathbf{y}_{LP} = \{\mathbf{y}_{LP}^{1}, \mathbf{y}_{LP}^{2}, ..., \mathbf{y}_{LP}^{N_n}\}$. Note that the i-th local patch observation vectors $\mathbf{y}_{LP}^{i}$ from previous tracking results are extracted and then used to generate the corresponding local PCA basis vectors $\mathbf{U}_{LP}^{i}$ using incremental subspace learning [1]. Figure 3.3 shows the block diagram to obtain the local PCA basis vectors $\mathbf{U}_{LP}^{1}$ corresponding to 1-st local patch using frames from 1 to 5. As the occlusion/outlier information for a given candidate sample $\mathbf{y}$ remains the same whether it is modeled by global or local appearance model, the global diagonal weight matrix $\mathbf{W}$ obtained during the IRRC is used to the reduce the effects of occlusion/outlier on the estimation of the local PCA basis coefficients $\mathbf{z}_{LP}$. Note that for a given local patch $\mathbf{y}_{LP}^{i}$ of the candidate sample $\mathbf{y}$, the respective local diagonal weight matrix $\mathbf{W}_{LP}^{i}$ is extracted from the global diagonal weight matrix $\mathbf{W}$. Now, the local PCA basis coefficients $\mathbf{z}_{LP}^{i}$ of the i-th local patch is obtained by

$$\mathbf{z}_{LP}^{i} = \left(\left(\mathbf{U}_{LP}^{i}\right)^{\mathsf{T}}\mathbf{W}_{LP}^{i}\mathbf{U}_{LP}^{i}\right)^{-1}\left(\mathbf{U}_{LP}^{i}\right)^{\mathsf{T}}\mathbf{W}_{LP}^{i}\,\overline{\mathbf{y}}_{LP}^{i} \tag{3.10}$$

where $\overline{\mathbf{y}}_{LP}^{i} = \mathbf{y}_{LP}^{i} - \boldsymbol{\mu}_{LP}^{i}$ is the centered i-th local image observation vector. Finally, the

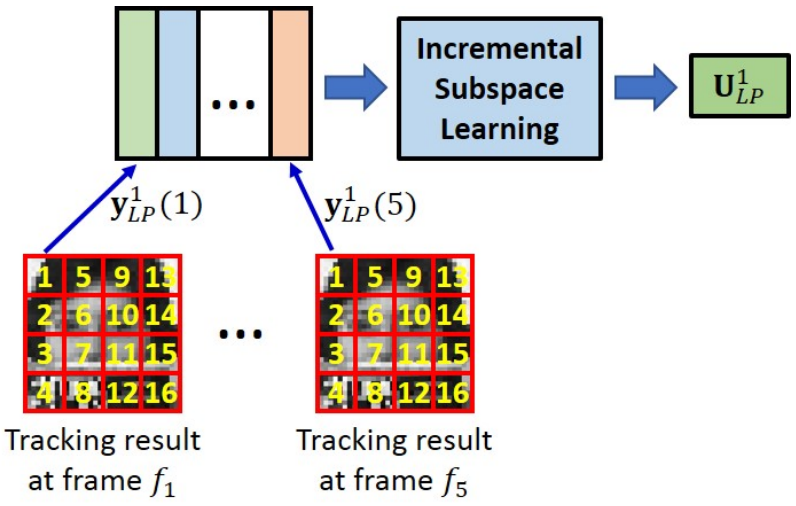


Figure 3.3: Block diagram to obtain local PCA basis vectors for the local model.

set of local PCA basis coefficients $\mathbf{z}_{LP} = \{\mathbf{z}_{LP}^{1}, \mathbf{z}_{LP}^{2}, ..., \mathbf{z}_{LP}^{N_n}\}$ for a given candidate $\mathbf{y}$ is obtained using the set of local diagonal weight matrices $\mathbf{W}_{LP} = \{\mathbf{W}_{LP}^{1}, \mathbf{W}_{LP}^{2}, ..., \mathbf{W}_{LP}^{N_n}\}$ as per Procedure 3.2.

Procedure 3.2 Computation of the set of local PCA basis coefficients $\mathbf{z}_{LP}$

Input: Set of centered local patch observation vectors $\overline{\mathbf{y}}_{LP}$, set of local PCA basis vectors $\mathbf{U}_{LP} = \{\mathbf{U}_{LP}^{1}, \mathbf{U}_{LP}^{2}, ..., \mathbf{U}_{LP}^{N_n}\}$, set of local diagonal weight matrices $\mathbf{W}_{LP}$, number of local patches N_n.

1: **for** $i = 1$ to N_n **do**

2: Compute $\mathbf{z}_{LP}^{i} = \left(\left(\mathbf{U}_{LP}^{i}\right)^{\mathsf{T}} \mathbf{W}_{LP}^{i} \mathbf{U}_{LP}^{i} \right)^{-1} \left(\mathbf{U}_{LP}^{i}\right)^{\mathsf{T}} \mathbf{W}_{LP}^{i} \overline{\mathbf{y}}_{LP}^{i}$

3: **end for**

Output: Set of local basis coefficients $\mathbf{z}_{LP}$.

Finally, the RC distance is derived to compare the candidate sample and the subspace in order to solve the tracking problem, which will be discussed in the next subsection.

3.2.4 Robust Coding Distance

Some of the vision applications, like visual tracking, not only require the accurate estimation of the coefficients, but also need a distance metric to find the similarity

between a noisy observation and its reconstructed sample from the dictionary or subspace [1, 65, 116]. In general, the distance metric is inversely proportional to the maximum joint likelihood with respect to the coefficient $\mathbf{z}$ [1, 65],

$$d\left(\mathbf{Y}; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}\right) \propto -\ln \max_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}) \propto -\ln \max_{\mathbf{z}} p(\mathbf{y}|\mathbf{z})\, p(\mathbf{z})$$

where $\mathbf{y} \in \mathbb{R}^{d_g \times 1}$ is a observation vector obtained from the observed image sample $\mathbf{Y} \in \mathbb{R}^{d_l \times d_r}$. Assuming an uniform prior, the distance metric is written as,

$$d\left(\mathbf{Y}; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}\right) \propto -\ln \max_{\mathbf{z}} \exp\left(-\frac{1}{2}\|\overline{\mathbf{y}} - \mathbf{U}_{GP}\,\mathbf{z}\|_2^2\right)$$

where $\overline{\mathbf{y}} = \mathbf{y} - \boldsymbol{\mu}_{GP}$ is the centered image observation vector. As this distance metric is a function of the global reconstruction error and is based on standard least squares (SLS), it is denoted as global SLS (GSLS) distance metric, and is given by

$$d_{GSLS}\left(\mathbf{Y}; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}\right) = \|\overline{\mathbf{y}} - \mathbf{U}_{GP}\,\mathbf{z}\|_2^2\,. \tag{3.11}$$

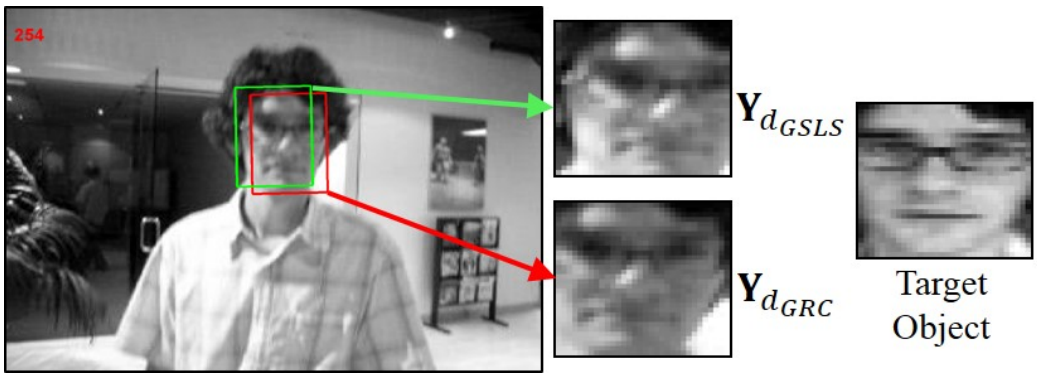
This SLS distance metric $d_{GSLS}\left(\mathbf{Y}; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}\right)$, which is sensitive to outliers/occlusion, is used in the likelihood function of the IVT [1] whose appearance model is also sensitive to outliers/occlusion, and hence the performance of IVT is not effective during occlusion. Note that in IVT, the basis coefficients are computed using $\mathbf{z} = \mathbf{U}_{GP}^{\mathsf{T}}\,\overline{\mathbf{y}}$. In order to make the distance metric robust to outliers/occlusion, the distance metric should give less importance to the reconstruction error due to occluded pixels or outliers, and hence, in the proposed algorithm [109], the distance between the observed image sample $\mathbf{Y}$ and the global PCA subspace $(\mathbf{U}_{GP}, \boldsymbol{\mu}_{GP})$ is defined as

$$d_{GRC}\left(\mathbf{Y}; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}\right) = \|\mathbf{W}^{\frac{1}{2}}\,\mathbf{e}\|_2^2 + \lambda_{GLP} \sum_j \left(1 - \mathbf{W}_{j,j}^{\frac{1}{2}}\right), \tag{3.12}$$

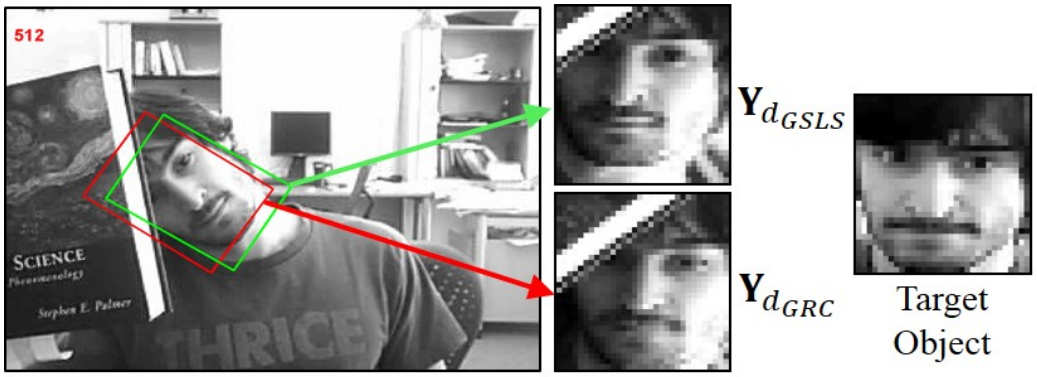
where, λ_{GLP} is a penalty constant. Due to the relation it has with RC, it is denoted as global RC distance. The first part of (3.12) accounts for the weighted reconstruction error of the candidate sample that reduces the influence of outliers/occlusion on the distance calculation compared to $d_{GSLS}\left(\mathbf{Y};\mathbf{U}_{GP},\boldsymbol{\mu}_{GP}\right)$ in (3.11), and the second part penalizes the labeling of any pixel as the outlier or being occluded. If the observed image sample $\mathbf{Y}$ can be represented well by the global PCA subspace $(\mathbf{U}_{GP},\boldsymbol{\mu}_{GP})$, the penalty term will be very small, otherwise very large. The importance of penalty term can be visualized when the candidate sample is from the background or dissimilar from the subspace, the residual will be large at all pixel locations leading to weights close to zero, and hence the distance metric may have lesser value due to the first term compared to the actual target sample, which makes the tracker to fail in the absence of penalty term. Similarly, the RC distance between the i-th local patch image $\mathbf{Y}^i_{LP}$ and i-th local PCA subspace $(\mathbf{U}^i_{LP},\boldsymbol{\mu}^i_{LP})$ is defined as

$$d_{LRC}\left(\mathbf{Y}^i_{LP};\mathbf{U}^i_{LP},\boldsymbol{\mu}^i_{LP}\right) = \|\left(\mathbf{W}^i_{LP}\right)^{\frac{1}{2}}\ \mathbf{e}^i_{LP}\|_2^2 +\ \lambda_{GLP}\sum_j\left(1-\left(\mathbf{W}^i_{LP}\right)^{\frac{1}{2}}_{j,j}\right), \tag{3.13}$$

where $\mathbf{e}^i_{LP}=\overline{\mathbf{y}}^i_{LP}-\mathbf{U}^i_{LP}\mathbf{z}^i_{LP}$ is i-th local patch error or residual vector. Figure 3.4 shows the candidates $\mathbf{Y}_{d_{GSLS}}$ and $\mathbf{Y}_{d_{GRC}}$ selected by the distance metrics d_{GSLS} and d_{GRC} respectively, for sequences having occlusion and motion blur, and their distances from the global PCA subspace $(\mathbf{U}_{GP},\boldsymbol{\mu}_{GP})$ using d_{GSLS}, d_{GSLS_RC} and d_{GRC} distance metrics are reported in Table 3.1, where the distance metric d_{GSLS_RC} is the SLS distance computed using (3.11) but with $\mathbf{z}=\mathbf{z}_{opt}$ obtained by the IRRC. It is observed from both Figure 3.4 and Table 3.1 that the proposed distance metric d_{GRC} finds the best match as compared to that of d_{GSLS} and d_{GSLS_RC} due to its occlusion/outlier handling capability when the object undergoes occlusion or motion blur. Also, for both the distance metrics d_{GSLS} and d_{GSLS_RC}, it is observed that $d_{GSLS}\left(\mathbf{Y}_{d_{GSLS}};\mathbf{U}_{GP},\boldsymbol{\mu}_{GP}\right) < d_{GSLS}\left(\mathbf{Y}_{d_{GRC}};\mathbf{U}_{GP},\boldsymbol{\mu}_{GP}\right)$ and $d_{GSLS\_RC}\left(\mathbf{Y}_{d_{GSLS}};\mathbf{U}_{GP},\boldsymbol{\mu}_{GP}\right) < d_{GSLS\_RC}\left(\mathbf{Y}_{d_{GRC}};\mathbf{U}_{GP},\boldsymbol{\mu}_{GP}\right)$, but for the proposed distance, $d_{GRC}\left(\mathbf{Y}_{d_{GSLS}};\mathbf{U}_{GP},\boldsymbol{\mu}_{GP}\right) > d_{GRC}\left(\mathbf{Y}_{d_{GRC}};\mathbf{U}_{GP},\boldsymbol{\mu}_{GP}\right)$.



(a) Motion Blur in *DavidIndoor*



(b) Occlusion in *Faceocc2*

Figure 3.4: Candidates selected by d_{GSLS} and d_{GRC}.

Even though the d_{GSLS_RC} uses the basis coefficients $\mathbf{z}_{opt}$, which are robust to outliers/occlusion, it fails to estimate the best match and performs similar to d_{GSLS}. This indicates that the outlier/occlusion handling capability should be effective both in the appearance model as well as in distance metric computation to achieve a better tracking performance.

3.3 Proposed Tracking Algorithm

The block diagram of the proposed tracking algorithm is shown in Figure 3.5. It is assumed that an observation vector $\mathbf{y}_t$ obtained from the image observation $\mathbf{Y}_t$ can be generated from a global PCA subspace of the target object spanned by the global

Table 3.1: Distance metrics comparison between the global PCA subspace ($\mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}$) and the candidates selected by d_{GSLS} and d_{GRC}. Lowest distances are in bold face, and the corresponding candidates are the best match.

Sequence/ Frame	Candidate	d_{GSLS}	d_{GSLS_RC}	d_{GRC}
***DavidIndoor* #254**	$\mathbf{Y}_{d_{GSLS}}$	26.79	28.23	6.35
	$\mathbf{Y}_{d_{GRC}}$	43.44	45.37	**3.85**
***Faceocc2* #512**	$\mathbf{Y}_{d_{GSLS}}$	44.20	50.73	6.94
	$\mathbf{Y}_{d_{GRC}}$	51.51	60.02	**2.75**

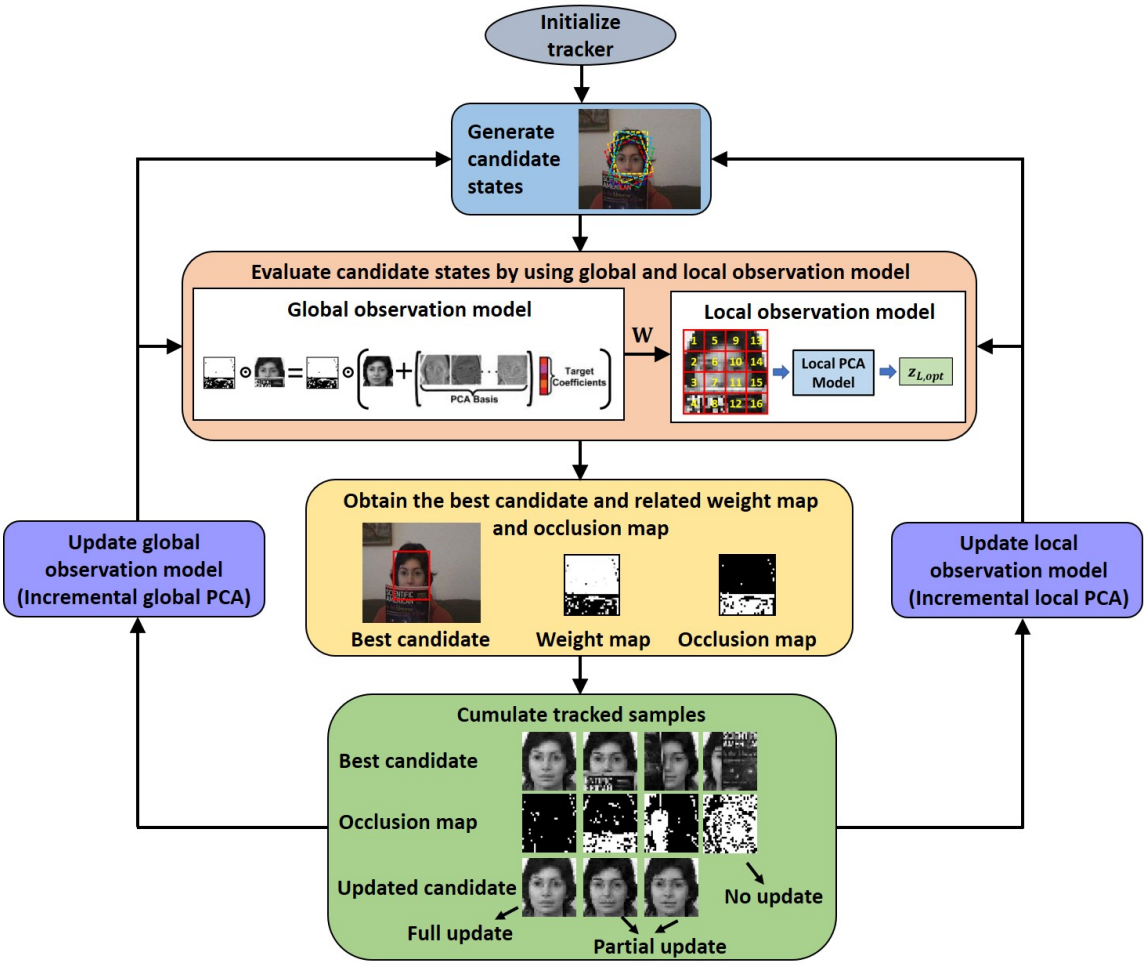


Figure 3.5: Block diagram of the proposed tracking algorithm.

PCA basis vectors $\mathbf{U}_{GP}$ and centered at global mean $\boldsymbol{\mu}_{GP}$. As opposed to the trivial templates used in other methods, the proposed algorithm uses the diagonal weight matrix to model the occlusion/outliers. Hence, even during occlusion, the weighted image observation can be represented by a weighted sum of the global mean $\boldsymbol{\mu}_{GP}$ and the linear combination of the global PCA basis vectors $\mathbf{U}_{GP}$ (as shown in Figure 3.1d), giving less importance to the outliers/occluded pixels in the representation as well as in the estimation of global basis coefficients $\mathbf{z}$. For each candidate vector $\mathbf{y}_t^m$ of a candidate image sample $\mathbf{Y}_t^m$ observed by a state $\mathbf{s}_t^m$, the minimization problem in (3.4) is solved efficiently by using the IRRC Procedure 3.1 to obtain $\mathbf{z}_t^m$ and $\mathbf{W}_t^m$. Similar to the global PCA model, it is assumed that the local patches of the same candidate image sample $\mathbf{Y}_t^m$ can be generated from the local PCA subspace (spanned by the local PCA basis vectors $\{\mathbf{U}_{LP}^i\}_{i=1}^{N_n}$, and centered at local mean $\{\boldsymbol{\mu}_{LP}^i\}_{i=1}^{N_n}$ of the corresponding patch). For every local patch $\mathbf{y}_{LP}^i$ of a candidate image sample $\mathbf{Y}_t^m$, the corresponding local diagonal weight matrix $\mathbf{W}_{LP}^i$ is extracted from the global diagonal weight matrix $\mathbf{W}_t^m$, and the local basis coefficients $\mathbf{z}_{LP}^i$ are computed as per Procedure 3.2. Now, the local RC distance between the candidate image sample $\mathbf{Y}_t^m$ and the local PCA subspace $(\mathbf{U}_{LP}, \boldsymbol{\mu}_{LP})$ is defined as

$$d_{LRC}\left(\mathbf{Y}_t^m; \mathbf{U}_{LP}, \boldsymbol{\mu}_{LP}\right) = \sum_{i=1}^{N_n} d_{LRC}\left(\mathbf{Y}_{LP}^i; \mathbf{U}_{LP}^i, \boldsymbol{\mu}_{LP}^i\right) \tag{3.14}$$

Further, the global and local models are collaborated by adding the global and local RC distances to get the final distance metric $d_{GLRC}\left(\mathbf{Y}_t^m; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}; \mathbf{U}_{LP}, \boldsymbol{\mu}_{LP}\right)$, which is defined as

$$d_{GLRC}\left(\mathbf{Y}_t^m; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}; \mathbf{U}_{LP}, \boldsymbol{\mu}_{LP}\right) = d_{GRC}\left(\mathbf{Y}_t^m; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}\right) + d_{LRC}\left(\mathbf{Y}_t^m; \mathbf{U}_{LP}, \boldsymbol{\mu}_{LP}\right)$$

In visual tracking based on particle filter framework, the confidence of the each particle is given by its observation likelihood, and in the proposed algorithm, a novel

observation likelihood is defined by

$$p(\mathbf{Y}_t^m|\mathbf{s}_t^m) = \exp\left(-\frac{1}{\gamma}\, d_{GLRC}\left(\mathbf{Y}_t^m; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}; \mathbf{U}_{LP}, \boldsymbol{\mu}_{LP}\right)\right) \tag{3.15}$$

where γ is a constant. As the proposed observation likelihood considers the distance metric $d_{GLRC}\left(\mathbf{Y}_t^m; \mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}; \mathbf{U}_{LP}, \boldsymbol{\mu}_{LP}\right)$, the effect of occlusion/outliers on the likelihood is reduced thereby making the likelihood robust to occlusion/outliers. Finally, the optimal state of the target $\hat{\mathbf{s}}_t$ is estimated using (2.4). Further, the global and local observation models are adapted to handle the appearance change of the target by incrementally updating the global ($\mathbf{U}_{GP}, \boldsymbol{\mu}_{GP}$) and local ($\mathbf{U}_{LP}, \boldsymbol{\mu}_{LP}$) PCA subspace models, as discussed in the next subsection.

3.3.1 Observation Model Update

For handling appearance variations of the target, the observation model update with precise samples is essential. In some scenarios of tracking, the occluders/outliers cover the target for an uncertain duration of time. The tracking result in these cases contain occlusion/outlier information in occluded part of the target and target information in non-occluded portion of the target. These tracking samples are not precise for observation model update due to occlusions/outliers. In order to remove the occlusion/outlier information, it is necessary to generate the occlusion map, which is essential to mask the occluded region during the observation model update. Otherwise, update with imprecise tracking samples with occlusions/outliers results in the model deterioration, thereby causing tracking failure. The observation model update with precise samples makes the model free from occlusions/outliers and hence the observation model could represent the candidate samples effectively in future frames. In the proposed algorithm, the occlusion/outlier information present in the tracked target candidate is captured by the residual error and the weight matrix $\hat{\mathbf{W}}$, which is obtained by the IRRC technique. Therefore, the weight matrix is exploited to detect

the occlusion/outliers as the lower weight values correspond to occlusion/outliers. The weight matrix $\hat{\mathbf{W}}$ is used to generate the occlusion map $\mathbf{O}$, which avoids the degradation of the observation model, and in the computation of the distance metric d_{GLRC}, which reduces the influence of occlusions/outliers on the tracking result, thereby improving the tracking results effectively. As each diagonal element of the weight matrix $\hat{\mathbf{W}}$ corresponds to a location in a 2D map, by placing the diagonal weight elements in respective positions (called as reverse raster scan) results in a weight map $\mathbf{W}_{map}$. Then, a binary occlusion map $\mathbf{O}$ is generated from the weight map $\mathbf{W}_{map}$ by assigning one and zero for the outlier and inlier pixels, respectively, using

$$\mathbf{O} = \begin{cases} 1, & \text{if } \mathbf{W}_{map} < 0.5 \\ 0, & \text{otherwise,} \end{cases} \tag{3.16}$$

Here, the weight $\mathbf{W}_{map} < 0.5$ is considered without any bias for the detection of outliers/occlusion as the weight values are bounded in [0,1] and 0.5 is midpoint of this range. Note that no arbitrary threshold is used to detect the occlusion/outlier as in other methods. Further, a occlusion ratio τ is computed as the ratio of the number of ones in the occlusion map to the total number of elements in the occlusion map $\mathbf{O}$. This occlusion ratio τ represents the amount of occlusion in the tracked target candidate and decides whether the update of the observation model with the tracked target candidate is full or partial or not, with the help of two thresholds τ_1 and τ_2. If $\tau < \tau_1$, then it is considered as the case of no occlusion and hence, the tracked target candidate will be used as it is for the model update (full update). If $\tau_1 < \tau < \tau_2$, then it is considered as the case of partial occlusion and hence, the occluded pixels in the tracked target candidate are replaced with the corresponding pixels from the previously updated mean $\boldsymbol{\mu}_{GP}$ to get a updated sample, which is free from occlusion, and is used in the model update. If $\tau > \tau_2$, then it is a case of severe occlusion and hence, this tracked target candidate will not be used in the model update. Some cases of these three update scenarios are shown in Figure 3.5.

After accumulating enough new updated tracked target candidates, which are free from occlusion, the global $(\mathbf{U}_{GP}, \boldsymbol{\mu}_{GP})$ and N_n local observation models $(\mathbf{U}_{LP}, \boldsymbol{\mu}_{LP})$ are updated with an incremental subspace learning [1]. Even though the update model of the proposed algorithm is similar to that of SPT [3], the difference lies in what is used to find the occlusion/outlier information. The proposed algorithm finds the occlusion information from the weights as the occluded pixels have low weights, whereas SPT does it from the coefficients of the trivial templates.

Algorithm 3.1 shows the steps of the proposed tracking method based on the global appearance model using RC and its collaboration with local model.

3.4 Experimental Results

Algorithm 3.1 is implemented in MATLAB and its average speed is 6.82 fps[1]. The algorithm when tested on gray scale features is denoted as Algorithm 3.1_Gray, and when tested on histogram of oriented gradient (HOG) features is denoted as Algorithm 3.1_HOG. Their performance is evaluated on the challenging sequences available in object tracking benchmark-50 (OTB-50) [7] and visual object tracking 2016 (VOT2016) [94] datasets using the respective evaluation protocols discussed in Section 2.3. In Algorithm 3.1, each image observation is resized to 32×32 pixels. This resized image is used for the global PCA representation as well as to extract $N_n = 16$ non-overlapped local patches of size 8×8 pixels for the local PCA representation while evaluating Algorithm 3.1_Gray. Similarly, to evaluate Algorithm 3.1_HOG, the cell size of 2×2 pixels with 5 orientations are adopted for extracting the HOG features from the resized image of 32×32 pixels. The above extracted HOG features are used for the global PCA representation, and for the local PCA representation by extracting $N_n = 16$ non-overlapped local patches of size 4×4. Further, 16 eigenvectors are used in all the experiments for both the global and local PCA subspace models. The

[1]Using modern computer of 3.4GHz CPU and 16GB RAM

Algorithm 3.1 Tracking algorithm based on the global appearance model using robust coding and its collaboration with the local model

Input: Target object is labeled in the first frame at $t = 1$ and its initial state is $\mathbf{s}_1$, number of PCA basis vectors k_g, number of non-overlapped local patches N_n, and constants c_1, c_2, τ_1 and τ_2.

Using the initial state $\mathbf{s}_1$, the target image $\mathbf{Y}_1$ is cropped out in the first frame and used to initialize both the global $\boldsymbol{\mu}_{GP}$ and local $\boldsymbol{\mu}_{LP}$ PCA mean vectors.
for $t > 1$ **do**
    Sample M candidate states $\{\mathbf{s}_t^1, \mathbf{s}_t^2, ..., \mathbf{s}_t^M\}$ from $\mathbf{s}_{t-1}$ using the particle filter.
    For every candidate state $\mathbf{s}_t^m$, extract the corresponding candidate sample $\mathbf{Y}_t^m$ and get the candidate vector $\mathbf{y}_t^m$, centered candidate vector $\overline{\mathbf{y}}_t^m$ and the set of centered local patch candidate vectors $\{\overline{\mathbf{y}}_{LP}^i\}_{i=1}^{N_n}$.
    if $t <= T_{GP}$ **then**
        if $t <= 5$ **then**
            For all the candidates $\mathbf{Y}_t^m$, compute d_{GRC} and d_{LRC} using (3.12) and (3.14), respectively, by assigning $\mathbf{W}_t^m = \mathbf{I}$, $\mathbf{e} = \overline{\mathbf{y}}_t^m$, and $\{\mathbf{e}_{LP}^i = \overline{\mathbf{y}}_{LP}^i\}_{i=1}^{N_n}$.
        else
            For all the candidates $\mathbf{Y}_t^m$, find d_{GRC} and d_{LRC} using (3.12) and (3.14), respectively, by assigning $\mathbf{W}_t^m = \mathbf{I}$, $\mathbf{e} = \overline{\mathbf{y}}_t^m - \mathbf{U}_{GP}\,\mathbf{U}_{GP}^{\top}\,\overline{\mathbf{y}}_t^m$, and $\{\mathbf{e}_{LP}^i = \overline{\mathbf{y}}_{LP}^i - \mathbf{U}_{LP}^i\,\left(\mathbf{U}_{LP}^i\right)^{\top}\,\overline{\mathbf{y}}_{LP}^i\}_{i=1}^{N_n}$.
        end if
        Find the optimal state of the tracked target $\hat{\mathbf{s}}_t$ using (3.15) and (2.4).
        The global $(\mathbf{U}_{GP}, \boldsymbol{\mu}_{GP})$ and N_n local $(\mathbf{U}_{LP}, \boldsymbol{\mu}_{LP})$ observation models are updated incrementally for every five frames as described in section 3.3.1 by assigning the global weight matrix $\mathbf{W}_t^m = \mathbf{I}$.
    else
        For all the candidate samples $\mathbf{Y}_t^m$, compute $\mathbf{z}_t^m$ and $\mathbf{W}_t^m$ as per Procedure 3.1.
        Extract the set of local diagonal weight matrices $\mathbf{W}_{LP}$ from the global weight matrix $\mathbf{W}_t^m$ and compute the set of local basis coefficients $\mathbf{z}_{LP}$ according to Procedure 3.2 for all the candidates.
        For all the candidates, calculate d_{GRC} and d_{LRC} using (3.12) and (3.14), respectively.
        Find the optimal state of the tracked target $\hat{\mathbf{s}}_t$ using (3.15) and (2.4).
        The global $(\mathbf{U}_{GP}, \boldsymbol{\mu}_{GP})$ and N_n local $(\mathbf{U}_{LP}, \boldsymbol{\mu}_{LP})$ observation models are updated incrementally for every five frames as described in section 3.3.1.
    end if
end for

Output: Target state $\hat{\mathbf{s}}_t$ at time t.

occlusion ratio thresholds τ_1 and τ_2 used to obtain occlusion-free samples in Section 3.3.1 are set to 0.1 and 0.6, respectively. Both the global and $N_n = 16$ local subspace models are updated after accumulating 5 occlusion-free samples using incremental subspace learning.

The computation of the two parameters β_{GP} and δ_{GP} needed in the weight function (3.5) during the residual minimization is explained now. In order to compute the value of β_{GP}, which controls the location of the demarcation point, it is assumed that when there are no occlusions in the tracked target candidate, the coding residuals $\hat{e_i}$ at all locations are in the "nominal range" and follow the Gaussian distribution. But during occlusion, the coding residuals $\hat{e_i}$ at the occluded locations will be probably above the "nominal range". Therefore, by finding the "nominal range" of the residuals in the starting frames ($t = 2$ to T_{GP}) of the given sequence, the value of the β_{GP} is calculated as

$$\beta_{GP} = \left(\frac{1}{T_{GP} - 1} \sum_{t=2}^{T_{GP}} \mathsf{mean}(\hat{\mathbf{e}}_t) + c_1 \, \mathsf{std}(\hat{\mathbf{e}}_t) \right)^2 \tag{3.17}$$

where c_1 is a constant, $\hat{\mathbf{e}}_t$ is the residual error of the tracked target candidate, and T_{GP} is the time instant at which the number of PCA basis vectors k_g becomes 16 for the first time. The first frame at $t = 1$ is manually labeled, due to which all the elements of the residual will be zero, and hence, the frame at $t = 1$ is not considered in the computation of β_{GP}. Now, the square of the residual $\hat{e_j}$ that is larger than the computed β_{GP}, will be considered as occluded/outlier and the value of weight will be less than 0.5. Further, the parameter δ_{GP}, which controls the decreasing rate of weight from 1 to 0 is computed using $\delta_{GP} = c_2/\beta_{GP}$, where c_2 is a constant. Here, the constants c_1 and c_2 are set as 2.5 and 6, respectively, for all sequences. In Algorithm 3.1, the IRRC technique will start functioning after the number of PCA basis vectors k_g becomes 16 for the first time. At time $t = T_{GP}$, the number of PCA basis vectors k_g is 16, and then the parameters β_{GP} and δ_{GP} are computed, which are then used by the IRRC technique for $t > T_{GP}$ to calculate the weights in (3.5).

The performance of Algorithm 3.1 is evaluated against several particle filter-based algorithms for a fair comparison. The algorithms considered are IVT [1], tracking based on L1 accelerated proximal gradient (L1APG) [13], SPT [3], weighted residual minimization in PCA subspace for visual tracking (WRMPCA) [110], locally weighted inverse sparse tracker (LWIST) [117], visual tracking via weighted local cosine similarity (WLCS) [108], visual tracking via least soft-threshold squares (LSST) [65], visual tracking via bilateral 2DPCA and robust coding (B2DPCARC) [118], robust object tracking via probability continuous outlier model (PCOM) [119] and deep learning tracker (DLT) [120]. The codes of the trackers IVT [1], L1APG [13], SPT [3], LWIST [117], WLCS [108], LSST [65], PCOM [119] and DLT [120] downloaded from the respective authors' website [121–127] and [128] are used to evaluate on the sequences of the benchmarks to have a fair comparison with Algorithm 3.1. The parameter settings of these trackers are as given in their respective papers in all the experiments, except for the parameters related to the particle filter framework, which are set to be as in OTB-50 [7] for a fair comparison. In the following two sub-sections, two benchmark datasets and the evaluation measures discussed in Section 2.3 are used for the quantitative evaluation of Algorithm 3.1 with that of the other methods.

3.4.1 Quantitative Evaluation on OTB-50

The performance of Algorithm 3.1 is evaluated on the OTB-50 benchmark dataset [7] and compared with that of the state-of-the-art tracking algorithms using one-pass evaluation (OPE). The performance comparison of Algorithm 3.1 in terms of the *precision score* for the location error threshold of 20 pixels with that of the other trackers for different attributes is given Table 3.2. The best three results are shown in **(red, bold)**, (violet, underline) and *(blue, italic)* fonts for better comparison of Algorithm 3.1 with the other trackers. It is observed that Algorithm 3.1_HOG provides the best performance in the face of all the challenging attributes, except occlusion (Occ) and deformation (Def), where it stood second, motion blur (MB)

and fast motion (FM), where it stood third, and low resolution (LR), where it stood seventh. On the other hand, Algorithm 3.1_Gray stands third in all the attributes, except scale variation (SV), MB, FM and in-plane rotation (IPR). Further, DLT ranks first in Occ, Def, MB and FM, and second in the remaining attributes except LR. L1APG stands second in MB, FM and LR, and third in IPR, whereas LSST stands first and third in LR and SV, respectively. It is also observed that Algorithm 3.1_HOG outperforms the other methods for most of the challenging attributes, except Occ, Def, MB, FM and LR.

Table 3.3 shows the performance comparison of Algorithm 3.1 in terms of the area under curve (AUC) with that of the state-of-the-art trackers for different challenging attributes. It is noticed that Algorithm 3.1_HOG provides the best performance in the face of all the challenging attributes except MB, where it stood second, and FM and LR, where it stood third. Further, Algorithm 3.1_Gray ranks second in LR, and third in IV, out-of-plane rotation (OPR), SV, Occ, Def and out-of-view (OV) challenging attributes. DLT stands first in MB and FM, second in illumination variation (IV), OPR, SV, Occ, Def, IPR and OV, and third in background clutter (BC) challenging attributes. L1APG stands second in FM and BC, and third in MB and IPR challenging attributes. Also, it is noticed that Algorithm 3.1_HOG outperforms the other methods for most of the challenging attributes, except MB, FM and LR.

In IVT, the appearance model and the likelihood function cannot handle outliers/occlusion effectively, resulting in a tracking drift for Occ and BC challenging attributes, and thus, the results are not satisfactory in these challenging attributes as well as in other attributes. On the other hand, the appearance model in WRMPCA handles the outliers/occlusion efficiently using RC and removes its effect on the computation of the basis coefficients, but does not remove its effect on the observation likelihood. Due to this, WRMPCA performs better than IVT does in all the challenging attributes, except in OV. In spite of using RC in B2DPCARC, the performance of

the B2DPCARC is inferior to that of IVT in all the challenging attributes, except in Def and FM, and WRMPCA in all the challenging attributes, except in Def. This is because the effect of outliers/occlusion is not considered while computing the observation likelihood of B2DPCARC. Even though both SPT and LSST are based on PCA subspace, their outlier/occlusion detection schemes are not so efficient as the one in Algorithm 3.1 and hence, their performance is inferior to that of Algorithm 3.1 in all the challenging attributes. In spite of using local patch information, both WLCS and LWIST do not capture the outliers/occlusion information in detail to the pixel level as compared to Algorithm 3.1 does, and hence, their performance is inferior to that of Algorithm 3.1. Further, the combined effect of using binary indicator vector (in contrast to the continuous valued weights used in Algorithm 3.1) to capture the information on occlusion/outliers and the absence of penalizing the pixel for labeling it as outlier or being occluded results in the poor performance of PCOM compared to that of Algorithm 3.1. Even though Algorithm 3.1_HOG has performed well in terms of AUC due to its outliers/occlusion handling capability, its performance is not that good in terms of the *precision score* for the challenging attributes Occ and Def. This may happen when the target undergoes several changes simultaneously along with severe occlusion. A similar argument holds good for the inferior performance of Algorithm 3.1_HOG, in terms of both the *precision score* and AUC, for the attributes MB, FM and LR, when the object undergoes a large appearance change due to fast motion of the object, and when there is a reduction in information of the object due to motion blur or low resolution.

The performance of Algorithm 3.1 is compared using the *precision* and *success plots* of OPE for OTB-50 benchmark sequences having occlusion against the other trackers and shown in Figure 3.6. In the *precision plot*, the *precision score* for the location error threshold of 20 pixels is used to rank the tracker, whereas in the *success plot*, AUC is used to rank the overall performance of the tracker. It is observed that Algorithm 3.1_HOG underperforms DLT by 2.7%, and outperforms L1APG,

Table 3.2: The *precision score* of Algorithm 3.1 with that of the compared trackers for different attributes of OTB-50. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Precision score	**IV**	**OPR**	**SV**	**Occ**	**Def**	**MB**	**FM**	**IPR**	**OV**	**BC**	**LR**
Algorithm 3.1_HOG	**56.5**	**56.3**	**60.3**	55.8	52.6	*36.5*	*35.1*	**58.3**	**51.0**	**57.4**	55.7
Algorithm 3.1_Gray	*42.8*	*50.4*	53.2	*52.3*	*49.5*	26.2	24.8	48.5	*36.3*	*44.8*	*60.2*
IVT	42.4	46.8	50.3	46.0	40.5	22.2	22.3	46.7	31.2	42.1	55.6
WRMPCA	42.7	47.7	52.8	45.9	41.2	24.3	23.6	47.5	31.1	43.4	58.7
B2DPCARC	30.7	42.0	41.7	40.0	43.4	19.9	23.2	41.4	27.6	38.2	56.1
L1APG	34.1	47.8	47.2	46.1	38.3	37.5	36.5	*51.8*	32.9	42.5	61.5
SPT	17.2	19.7	23.6	18.9	18.2	10.3	13.5	22.9	17.4	20.5	21.4
WLCS	32.9	33.0	38.3	35.7	29.9	15.4	21.3	30.3	22.8	33.1	23.0
LWIST	34.3	38.4	40.3	40.4	35.3	15.1	18.7	33.7	30.7	37.4	45.9
PCOM	36.8	46.1	48.6	44.4	38.3	22.9	22.0	46.0	30.5	40.1	55.9
DLT	53.4	56.1	59.0	**57.4**	**56.3**	**45.3**	**44.6**	54.8	44.4	49.5	53.6
LSST	41.5	47.0	*53.3*	41.3	45.0	27.1	25.2	46.1	24.3	42.6	**74.1**

Table 3.3: AUC of Algorithm 3.1 with that of the compared trackers for different attributes of OTB-50. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

AUC	**IV**	**OPR**	**SV**	**Occ**	**Def**	**MB**	**FM**	**IPR**	**OV**	**BC**	**LR**
Algorithm 3.1_HOG	**46.6**	**44.6**	**48.2**	**44.0**	**42.0**	31.6	*30.2*	**46.4**	**43.2**	**46.4**	*41.6*
Algorithm 3.1_Gray	*33.7*	*37.0*	*38.6*	*37.7*	*34.6*	23.3	22.3	37.1	*32.3*	31.3	41.7
IVT	30.9	32.7	34.8	32.7	28.4	19.4	20.2	33.4	27.7	29.5	34.2
WRMPCA	33.2	34.7	37.4	33.8	29.7	20.8	21.6	35.1	27.4	31.6	38.9
B2DPCARC	23.9	29.2	30.2	29.6	31.2	18.4	20.8	28.8	26.0	26.8	35.8
L1APG	28.3	36.0	35.0	35.3	31.1	*31.0*	31.1	*39.1*	30.3	35.0	37.4
SPT	16.6	17.1	18.3	16.7	13.4	11.9	13.7	20.1	17.7	16.0	14.0
WLCS	27.0	25.8	29.5	27.6	25.0	15.8	19.6	24.2	21.6	27.3	13.7
LWIST	28.9	30.1	31.3	31.9	28.5	15.1	18.3	27.2	28.1	31.1	27.8
PCOM	28.4	32.3	32.9	33.2	26.0	17.8	18.8	32.4	27.5	28.4	34.5
DLT	40.5	41.2	45.5	42.3	39.4	**36.3**	**36.0**	41.1	36.7	*33.9*	34.7
LSST	33.2	33.5	37.7	30.4	31.6	21.4	21.5	32.3	21.7	30.2	**46.2**

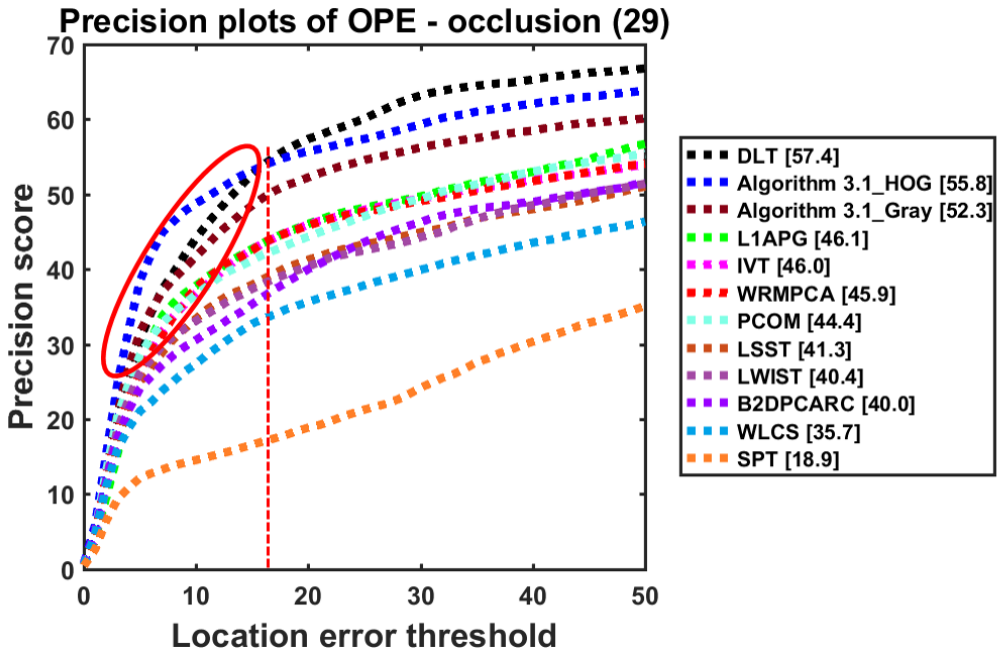
IVT, WRMPCA, PCOM, LSST, LWIST, B2DPCARC and WLCS by 21.0%, 21.3%, 21.5%, 25.7%, 35.1%, 38.1%, 39.5% and 56.3%, respectively, in terms of the *precision score*. Even though Algorithm 3.1_HOG underperforms DLT in terms of the *precision score*, which is evaluated at the location error threshold of 20 pixels, it does better than that of the latter for the location error threshold < 15. On the other hand, Algorithm 3.1_HOG outperforms DLT, L1APG, WRMPCA, PCOM, IVT, LWIST, LSST, B2DPCARC and WLCS by 4.0%, 24.6%, 30.1%, 32.5%, 34.5%, 37.9%, 44.7%, 48.6% and 59.4%, respectively, in terms of AUC. Also, it is observed that the *success rate* of Algorithm 3.1_HOG is less than that of DLT for overlap threshold < 0.4 but it is better than that of the latter when the overlap threshold > 0.4. Note that for a best tracking performance, the *precision score* should be high even for a lower values of the location error threshold and the *success rate* should be high even for a higher values of the overlap threshold. Therefore, the performance of Algorithm 3.1 is better than that of the other methods in terms of both the *success rate* and the *precision score*.

The *success* and *precision plots* of OPE for the considered trackers averaging over the OTB-50 benchmark sequences are shown in Figure 3.7. It is observed from Figure 3.7 that Algorithm 3.1_HOG outperforms the state-of-the-trackers DLT, WRMPCA, IVT, LSST, L1APG, PCOM, B2DPCARC, LWIST and WLCS by 3.9%, 18.9%, 21.2%, 23.4%, 25.7%, 28.1%, 40.2%, 49.1% and 68.5%, respectively, in terms of the *precision score*. Further, Algorithm 3.1_gray underperforms DLT by 6.3%, and outperforms WRMPCA, IVT, LSST, L1APG, PCOM, B2DPCARC, LWIST and WLCS by 7.2%, 9.3%, 11.3%, 13.4%, 15.5%, 26.4%, 34.5% and 51.9%, respectively, in terms of the *precision score*. Also, it is observed that Algorithm 3.1_HOG performs best for the location error threshold < 30 in terms of the *precision score*. Similarly, Algorithm 3.1_HOG outperforms DLT, WRMPCA, L1APG, IVT, LSST, PCOM, LWIST, B2DPCARC and WLCS by 11.2%, 27.3%, 27.6%, 34.7%, 36.2%, 42.6%, 49.2%, 58.5% and 69.5%, respectively, in terms of AUC. Also, Algorithm 3.1_gray underperforms

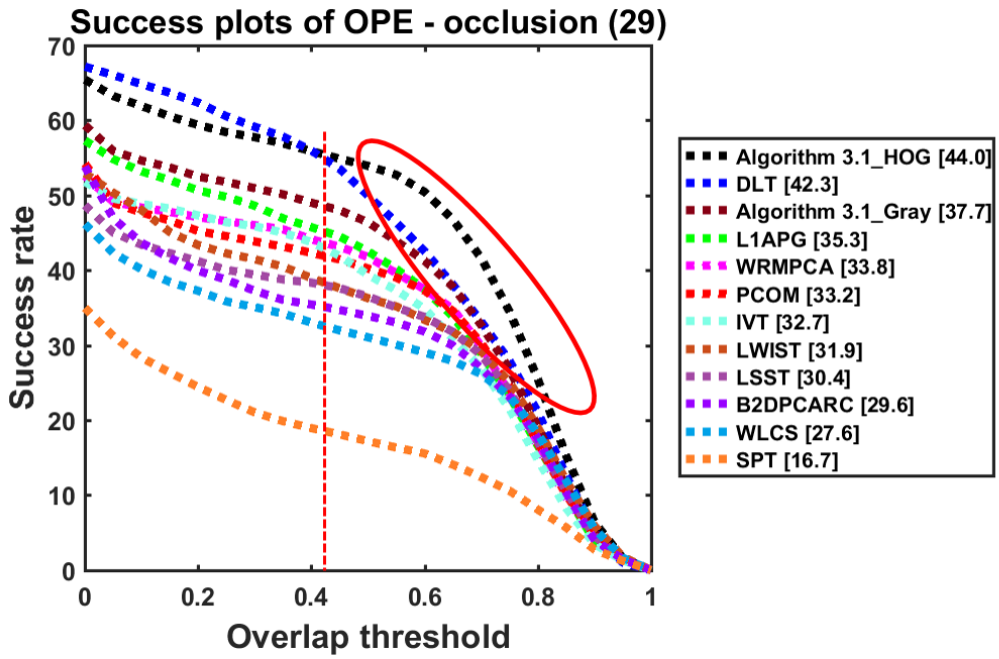
DLT by 7.1%, and outperforms WRMPCA, L1APG, IVT, LSST, PCOM, LWIST, B2DPCARC and WLCS by 6.3%, 6.6%, 12.5%, 13.7%, 19.1%, 24.6%, 32.3% and 41.6%, respectively. Further, the best performance of the Algorithm 3.1_HOG can be observed for overlap threshold > 0.2 in terms of the *success rate*. Overall, Algorithm 3.1_HOG provides the best performance to that of the other methods in terms of both the *precision score* and the *success rate*.

3.4.2 Quantitative Evaluation on VOT2016

The average accuracy and robustness, and their rank are used to evaluate the performance of Algorithm 3.1 using VOT2016 benchmark dataset [94]. Table 3.4 shows the accuracy rank and overlap comparison of Algorithm 3.1 with that of the state-of-the-art tracking algorithms averaging over the VOT2016 sequences. Similarly, Table 3.5 shows the performance comparison of Algorithm 3.1 using robustness rank and failures averaging over the same challenging sequences. The last six columns of these two tables show the respective measures using different averaging methodologies, mean, weighted mean and pooled. The best three results are shown in **(red, bold)**, (violet, underline) and *(blue, italic)* fonts for better comparison of the proposed tracker with the other state-of-the-art trackers. Note that as the trackers with statistically equivalent results are merged while ranking, the different trackers may have same accuracy rank and robustness rank [94]. It is observed from Table 3.4 that Algorithm 3.1_HOG ranks first in all the attributes and averages of attributes (mean, weighted mean and pooled), and stands third for the attributes illumination change and motion change. On the other hand, Algorithm 3.1_Gray stands second and third for the attribute motion change, and for the attributes camera motion, size change, weighted mean and pooled, respectively. Also, DLT ranks first and second for the attribute motion change, and for the attributes camera motion, empty, occlusion, size change, mean, weighted mean and pooled, respectively. Further, LWIST stands first and third for the attributes illumination change and mean, respectively, whereas

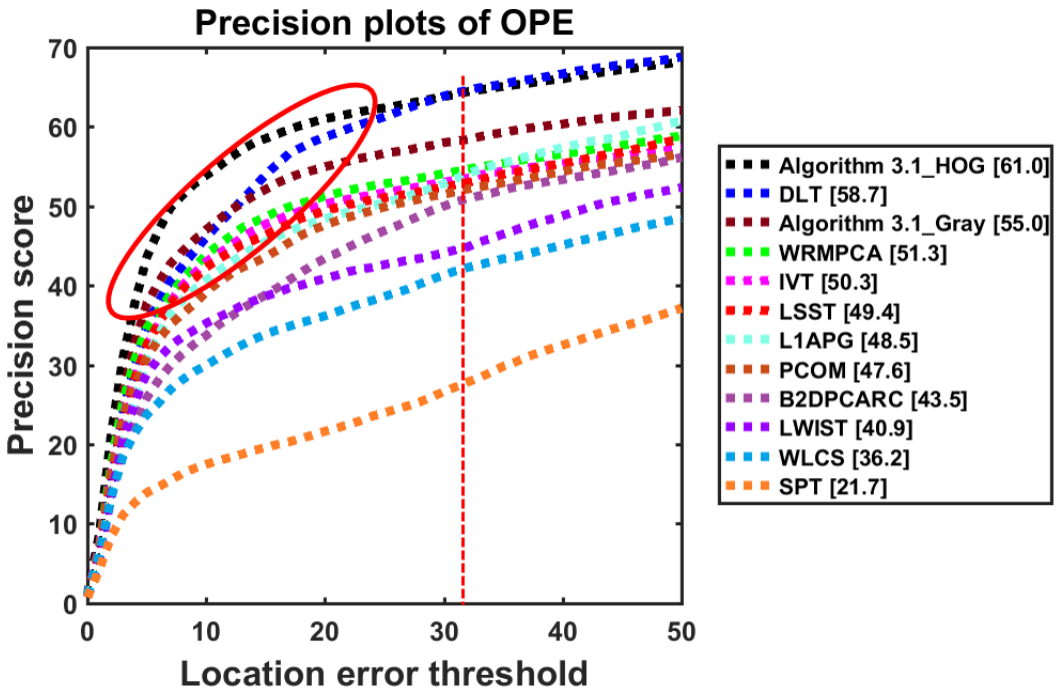


(a)

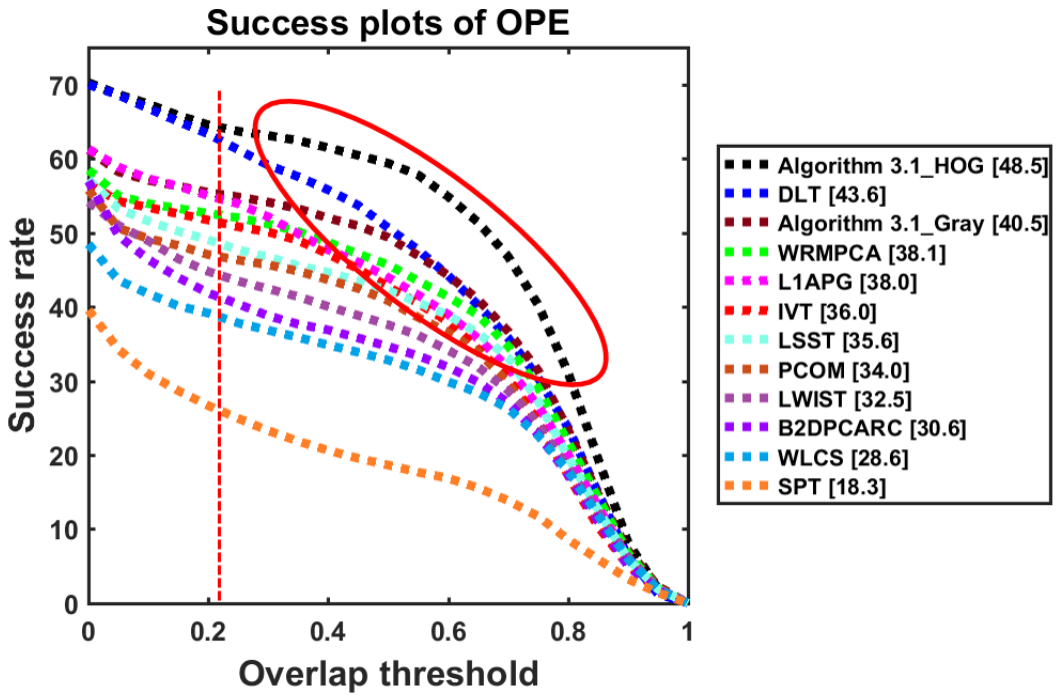


(b)

Figure 3.6: Performance evaluation of Algorithm 3.1 on OTB-50 sequences having occlusion using (a) the *precision plots* of OPE, where the *precision score* is shown along with the tracker name in the legend, and (b) the *success plots* of OPE, where AUC is shown along with the tracker name in the legend. The number 29 appearing in the title denotes the number of sequences associated with the occlusion attribute of OTB-50 dataset.



(a)



(b)

Figure 3.7: Overall performance evaluation of Algorithm 3.1 on OTB-50 using (a) the *precision plots* of OPE, where the *precision score* is shown along with the tracker name in the legend, and (b) the *success plots* of OPE, where AUC is shown along with the tracker name in the legend.

WLCS stands second and third for the attributes illumination change and empty, respectively. Also, PCOM ranks third for the attribute occlusion. Further, Algorithm 3.1_HOG performs superior to that of the other methods for all the challenging attributes except illumination change and motion change.

From Table 3.5, it is observed that Algorithm 3.1_HOG stands first for the attribute size change, and third for motion change and occlusion. On the other hand, Algorithm 3.1_Gray ranks second for the attributes motion change, occlusion, mean, weighted mean and pooled, and third for camera motion and empty. Further, DLT stands first for all the attributes except size change, where it stood second and illumination change, where it stood seventh. WRMPCA stands second for camera motion, and third for size change, mean, weighted mean and pooled. Also, LWIST and WLCS ranks first and third for illumination change, respectively, and L1APG ranks second for the attribute empty. Even though Algorithm 3.1_HOG is inferior to DLT in terms of robustness on most of the challenging attributes, it performs better than the latter does in terms of accuracy. This may happen when the target undergoes severe occlusion, scale change, out-of-plane rotation, motion blur, fast motion either individually or simultaneously.

3.4.3 Qualitative Evaluation

For qualitative evaluation of the trackers, some tracking results on a subset of the OTB-50 benchmark sequences are shown in Figure 3.8. For each sequence, the tracking results of all the trackers on the six exemplar image frames, which are selected at regular intervals without any bias, are shown. Algorithm 3.1_HOG successfully tracks the target in the all the frames of all the sequences, which contain most of the real-time challenges such as pose changes, partial occlusion, out-of-plane rotation, illumination and scale variations. It is also observed that Algorithm 3.1_HOG is the only method that tracks the object successfully in *Suv* and *Trellis* sequences. This shows that Algorithm 3.1 is strong enough to handle these challenges. Further,

Table 3.4: Accuracy rank (A-Rank) and average overlap comparison of Algorithm 3.1 with that of the compared trackers for different attributes of VOT2016. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Accuracy	camera motion		empty		illumination change		motion change		occlusion		size change		Mean		Weighted mean		Pooled	
Trackers	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap
Algorithm 3.1_HOG	**1.00**	**0.45**	**1.00**	**0.52**	**1.00**	*0.61*	**1.00**	*0.38*	**1.00**	**0.38**	**1.00**	**0.43**	**1.00**	**0.46**	**1.00**	**0.45**	**1.00**	**0.46**
Algorithm 3.1_Gray	**1.00**	*0.43*	**1.00**	0.48	4.00	0.54	**1.00**	0.38	**1.00**	0.34	**1.00**	*0.41*	*1.50*	0.43	*1.50*	*0.43*	**1.00**	*0.43*
IVT	**1.00**	0.41	**1.00**	0.48	5.00	0.50	**1.00**	0.36	**1.00**	0.33	**1.00**	0.37	1.67	0.41	1.67	0.41	**1.00**	0.42
WRMPCA	**1.00**	0.40	**1.00**	0.49	4.00	0.52	**1.00**	0.37	**1.00**	0.31	**1.00**	0.37	*1.50*	0.41	*1.50*	0.41	**1.00**	0.42
SPT	12.00	0.31	11.00	0.36	9.00	0.45	*7.00*	0.27	3.00	0.29	*6.00*	0.31	8.00	0.33	8.00	0.32	11.00	0.32
B2DPCARC	**1.00**	0.42	**1.00**	0.47	*3.00*	0.54	**1.00**	0.36	**1.00**	0.35	**1.00**	0.37	1.33	0.42	1.33	0.41	**1.00**	0.42
WLCS	**1.00**	0.41	**1.00**	*0.50*	**1.00**	0.63	**1.00**	0.34	**1.00**	0.33	**1.00**	0.40	**1.00**	0.43	**1.00**	0.42	**1.00**	0.43
LWIST	**1.00**	0.42	**1.00**	0.49	**1.00**	**0.65**	**1.00**	0.34	**1.00**	0.34	**1.00**	0.41	**1.00**	*0.44*	**1.00**	0.42	**1.00**	0.43
PCOM	**1.00**	0.41	**1.00**	0.49	*3.00*	0.57	**1.00**	0.36	**1.00**	*0.36*	**1.00**	0.41	1.33	0.43	1.33	0.42	**1.00**	0.43
DLT	**1.00**	0.44	**1.00**	0.50	**1.00**	0.58	**1.00**	**0.39**	**1.00**	0.38	**1.00**	0.42	**1.00**	0.45	**1.00**	0.44	**1.00**	0.45
LSST	**1.00**	0.40	**1.00**	0.44	2.00	0.55	2.00	0.32	**1.00**	0.34	**1.00**	0.37	1.33	0.40	1.33	0.39	**1.00**	0.40
L1APG	**1.00**	0.43	**1.00**	0.50	10.00	0.45	**1.00**	0.37	**1.00**	0.35	3.00	0.33	2.83	0.40	2.83	0.41	**1.00**	0.42

Table 3.5: Robustness rank (R-Rank) and average failures comparison of Algorithm 3.1 with that of the compared trackers for different attributes of VOT2016. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Robustness	camera motion		empty		illumination change		motion change		occlusion		size change		Mean		Weighted mean		Pooled	
Trackers	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures
Algorithm 3.1_HOG	**1.00**	109.27	*3.00*	62.53	**1.00**	10.00	2.00	*91.40*	**1.00**	*36.20*	**1.00**	**42.73**	1.50	58.69	1.50	73.54	2.00	248.00
Algorithm 3.1_Gray	**1.00**	*107.47*	*3.00*	*50.67*	**1.00**	9.47	2.00	88.73	**1.00**	36.13	**1.00**	48.60	1.50	56.84	1.50	70.60	2.00	238.93
IVT	2.00	114.67	*3.00*	59.40	**1.00**	9.47	2.00	101.87	2.00	40.00	3.00	52.47	2.17	62.98	2.17	78.10	2.00	264.47
WRMPCA	2.00	106.60	2.00	52.47	**1.00**	10.47	2.00	91.93	2.00	38.20	**1.00**	*48.33*	*1.67*	*58.00*	*1.67*	*71.51*	2.00	*242.60*
SPT	12.00	183.27	12.00	107.27	*9.00*	12.33	12.00	135.27	*4.00*	54.07	12.00	72.60	10.17	94.13	10.17	120.29	12.00	414.47
B2DPCARC	6.00	126.93	*3.00*	62.20	3.00	10.67	5.00	100.67	7.00	43.93	8.00	59.20	5.33	67.27	5.33	83.66	7.00	278.73
WLCS	7.00	164.73	11.00	89.53	**1.00**	*9.13*	9.00	124.13	*4.00*	46.67	8.00	60.87	6.67	82.51	6.67	105.86	11.00	355.27
LWIST	6.00	131.27	9.00	81.13	**1.00**	**8.53**	5.00	115.13	2.00	43.07	*5.00*	55.27	4.67	72.40	4.67	91.28	8.00	307.53
PCOM	*5.00*	126.80	*3.00*	57.67	**1.00**	8.87	*4.00*	102.27	*4.00*	41.20	3.00	53.60	3.33	65.07	3.33	81.51	*5.00*	273.67
DLT	**1.00**	**94.73**	**1.00**	**42.20**	**1.00**	10.07	**1.00**	**66.53**	**1.00**	**35.40**	**1.00**	43.33	**1.00**	**48.71**	**1.00**	**60.06**	**1.00**	**201.47**
LSST	7.00	135.60	9.00	77.87	3.00	11.40	5.00	115.20	2.00	45.60	9.00	61.47	5.83	74.52	5.83	93.11	9.00	315.20
L1APG	2.00	140.80	**1.00**	49.13	3.00	11.60	5.00	99.13	**1.00**	38.60	3.00	54.27	2.50	65.59	2.50	82.89	2.00	267.27

Algorithm 3.1_HOG has slightly drifted away in the last few frames of the *Fleetface* sequences shown in Figure 3.8. On the other hand, Algorithm 3.1_Gray fails to track the object in most of the sequences except *Faceocc2*, *Fleetface*, *Freeman3*, *David* and *Walking*. Algorithm 3.1 performs well for the challenges that are considerably difficult, however, loses the target for challenges that are extremely difficult or for the sequences where the target undergoes several changes simultaneously. It is also observed that LWIST and SPT fail in most of the sequences, whereas WLCS fails in all the sequences except *Faceocc2*. WRMPCA tracks the object successfully in *Faceocc2* and *Walking* sequences, fails to estimate the scale in *David*, fails to estimate both the scale and location in *Freeman3*, and fails in the remaining sequences. Further, B2DPCARC successfully tracks the object in *Doll* and *Walking* sequences, fails to estimate the scale in *Faceocc2* and *Singer2*, both scale and location in *Freeman3* and *David* and fails in the remaining sequences. Even though LSST tracks the object in *Singer2* sequence completely but fails to estimate the scale and location of the object accurately. Similar observations can be made for LSST in *Faceocc2* and *David* sequences with imprecise estimation of the scale and location of the object. Further, LSST and DLT have successfully tracked the object in *Freeman3* sequence. Also, DLT has tracked the object in *Faceocc2*, *Fleetface*, *CarScale* and *David* sequences with inaccurate scale and location of the object. On the other hand, PCOM fails to track the object in most of the frames in *Fleeface*, *Suv*, *Trellis* and *Walking* sequences, except in *Freeman3*, *CarScale* and *David*, where it fails to estimate the scale of the object. Also, IVT fails to track the object in most of the sequences except in *Faceocc2*, *CarScale* and *David* with imprecise estimation of scale. Thus, from these qualitative analyses, it is observed that Algorithm 3.1 performs favorably in most of the challenging sequences.

The main drawback of Algorithm 3.1 is that it fails to track the object when there is a large change in the appearance of the object. This is due to the fact that the reconstruction of the object from the linear PCA subspace may not be accurate as

the candidate samples may lie beyond the range of the linear subspace.

3.5 Summary

In this chapter, a new generative tracking algorithm, Algorithm 3.1, which is based on the global appearance model using robust coding and its collaboration with a local model has been proposed. The global PCA subspace has been used to model the global appearance of the object and an iteratively reweighted robust coding technique has been developed to compute the optimum global PCA basis coefficients and the global weight matrix. A collaborative scheme of the global and local appearance models has been presented to exploit their individual merits. The global and local robust coding distances have been introduced to find the candidate having the similar appearance as that of the reconstructed sample from the subspace, and the observation likelihood has been defined using the above two distances. The occlusion map has been generated and used to reduce the effect of outliers/occlusion on the observation model update. Extensive experiments have been conducted on the OTB-50 and VOT2016 benchmark datasets to analyze the performance of Algorithm 3.1. Also, Algorithm 3.1 has been tested on the gray values (Algorithm 3.1_Gray) and HOG features (Algorithm 3.1_HOG), and the corresponding performances using one-pass evaluation have been compared with the several state-of-the-art methods based on particle filter framework. Even though Algorithm 3.1_HOG generally performs better than the other methods do for most of the challenging attributes, there is still a room for improvement in occlusion, deformation, motion blur, fast motion and low resolution challenging attributes of OTB-50, and camera motion, illumination change, motion change and occlusion attributes of VOT2016. In order to improve the performance of the tracker on some of these challenging attributes, a second tracking algorithm based on a structural local 2DDCT sparse appearance model and an occlusion handling mechanism is introduced in the next chapter.

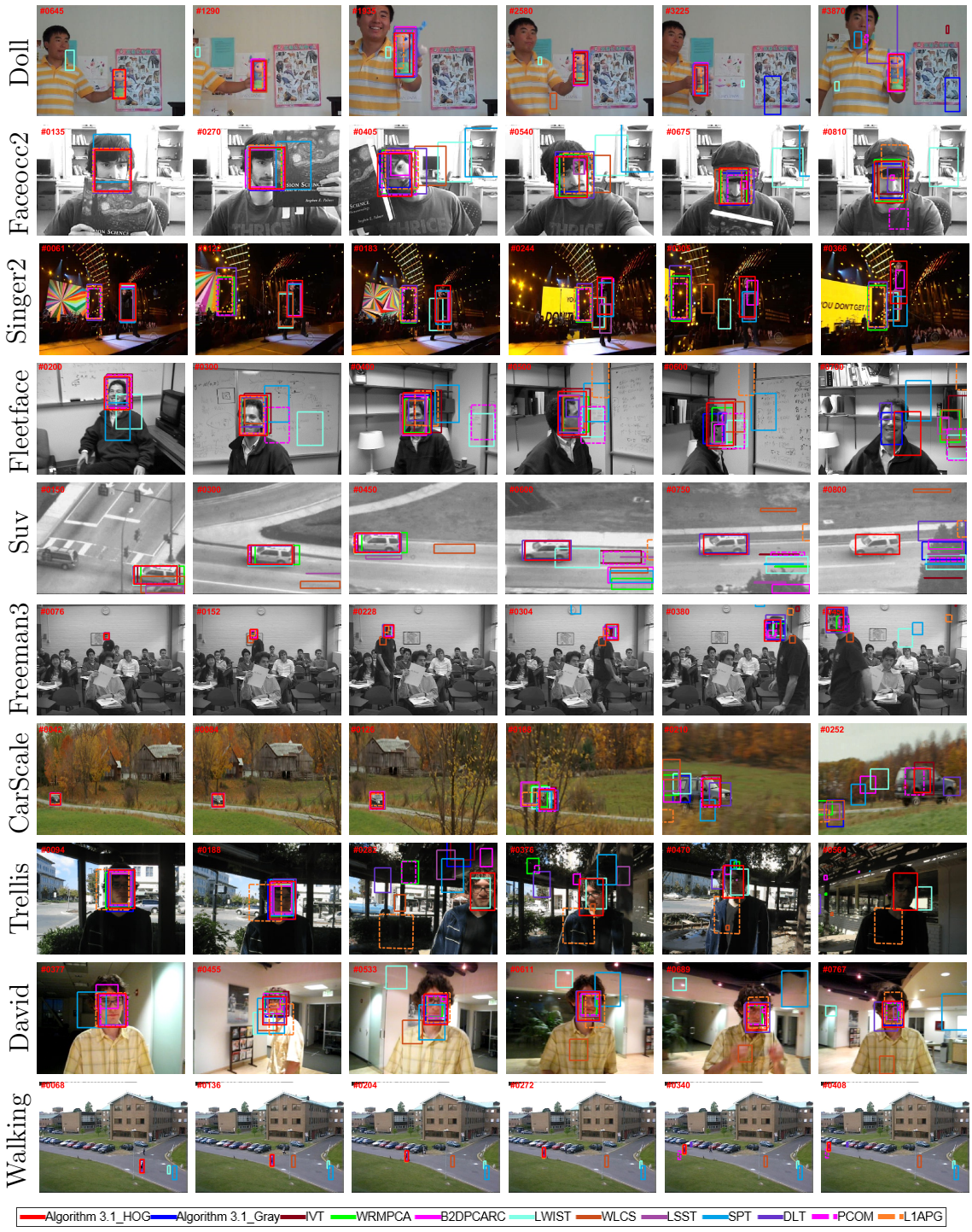


Figure 3.8: Examples of tracking results of the compared methods on the ten OTB-50 benchmark sequences.

Chapter 4

Structural Local 2DDCT Sparse Appearance Model and an Occlusion Handling Mechanism for Visual Tracking

4.1 Introduction

Algorithm 3.1_HOG [109], proposed in the previous chapter, does not perform well for the challenging attributes of occlusion (Occ), deformation (Def), motion blur (MB), fast motion (FM) and low resolution (LR) present in the object tracking benchmark-50 (OTB-50) dataset, and camera motion, illumination change, motion change and occlusion in the visual object tracking 2016 (VOT2016) benchmark dataset. This may happen when the target undergoes severe occlusion, deformation, out-of-plane rotation, illumination change, fast motion and appearance change due to camera motion either individually or simultaneously resulting in a large appearance variation of the object. When there is a large change in the appearance of the object, the reconstruction of the object from the principal component analysis (PCA) subspace model may not be accurate as the candidate samples may lie beyond the range of the PCA subspace. This problem is addressed by using a set of target templates to model the object appearance. Most of the trackers based on target templates

[2, 50, 52, 53, 129] use a holistic/global object representation scheme to handle the appearance variation of the object. However, these trackers fail during occlusions and background clutter as the information based on the parts of the target and spatial information within the target region are not fully exploited. On the other hand, the local appearance models [11, 15, 20, 108, 130] exploit the partial and spatial structural information of the target, and hence, perform better during occlusions and background clutter. Therefore, in this chapter, local patches that are extracted from the target templates in a structured manner are investigated in a transform domain for the object appearance model in visual tracking.

Adam *et al.* [11] proposed a tracking algorithm based on fragments, where each fragment is tracked by measuring the local regional similarity, and then, the target location is found by using the vote maps of the tracked fragments. Liu *et al.* [130] proposed a local sparse appearance model based on a static sparse dictionary and a dynamically updated basis distribution, and then tracking is achieved using a sparse representation-based vote map and a sparse regularized mean-shift algorithm. Jia *et al.* [15] proposed a visual tracking algorithm based on an adaptive structural local sparse appearance (ASLA) model by exploiting both the partial and the spatial information of the target. Similar to ASLA, Dai *et al.* [131] proposed a part-based sparsity model for visual tracking but with non-overlapped patches to model the object appearance. Then, the target template set for each patch is updated dynamically. Further, a tracking algorithm based on support vector machine is proposed by exploiting the aligned structural local sparse features in [30]. Concurrently, a robust local sparse tracker with global consistency constraint is proposed in [132] to alleviate the problem of drifting when a patch on the background is similar to some patches of the target. Wang *et al.* [108] proposed a weighted local cosine similarity (WLCS) to measure the similarity between the target and the candidates, and then developed a tracking algorithm based on the local model. Further, the discriminative ability of WLCS is improved by learning discriminative weights via quadratic programming.

Also, they proposed a locally weighted distance metric-based inverse sparse tracking algorithm (LWIST) [117], where instead of using the Euclidean distance metric, a locally weighted distance metric is employed to measure the similarity between the target and the candidates. As the algorithm employs a local template update scheme, the unoccluded local parts are updated while the occluded ones are discarded during heavy occlusion.

In this chapter, a new tracking algorithm [133] based on a structural local 2DDCT sparse appearance model [134] and an occlusion handling mechanism is proposed. The energy compaction property of 2DDCT is exploited by using only a few 2DDCT coefficients in the object appearance model to reduce the computational cost of the l_1-minimization. This strategy is unlike other models that use raw pixels for object representation. As raw pixels are equally probable, the dimensionality reduction in the spatial domain is not affordable. A method is presented to reconstruct a holistic image from the overlapped local patches that are obtained using the local patch dictionary and the sparse codes. A robust occlusion map generation scheme using the reconstructed holistic image, and the pooled feature vector is developed. The highest confident occlusion-free sample among the cumulated samples is used to reconstruct the image for the template update. The template that contributes least in representing the previous tracking results is replaced with the reconstructed image obtained after incremental subspace learning. A patch occlusion ratio is used while computing the confidence of a candidate. Experiments conducted on the two popular benchmark datasets with comparison to the state-of-the-art tracking methods bear out the competency and effectiveness of the proposed algorithm for visual tracking.

This chapter is organized as follows. Section 4.2 introduces the object representation using a structural local 2DDCT sparse appearance model. The holistic image reconstruction from an overlapped local patches is explained in Section 4.3 followed by the proposed tracking algorithm in Section 4.4. Experimental results for the two popular benchmark datasets, OTB-50 and VOT2016, are presented and discussed in

Section 4.5 followed by a chapter summary in Section 4.6.

4.2 Structural Local 2DDCT Sparse Appearance Model

In the literature, only a few attempts have been made to exploit the properties of 2DDCT for visual tracking [36, 135, 136] in spite of its success in a wide range of vision applications such as image retrieval [137, 138], image fusion [139, 140], image denoising [141, 142], face recognition [143, 144], video object segmentation [145] and video caption localization [146]. In both [135] and [136], the features extracted from the 2DDCT coefficients are used to find the target location by measuring the similarity between the target and the candidates, but there is no update of appearance model in [135]. At the same time, Li *et al.* [36] proposed a compact 3DDCT based object representation and its incremental learning for robust visual tracking (IL3DDCT) using a signal reconstruction-based similarity measure to evaluate the likelihood. In contrast to these methods, which use only 2DDCT for the appearance model, the proposed appearance model [134] exploits both the sparse representation and 2DDCT to model the appearance of the object. Further, the proposed appearance model uses the local patches in contrast to the holistic templates used in the above methods. Also, a mechanism for robust occlusion detection and observation model update mechanisms are developed to reduce the effects of occlusion/outliers on the tracking algorithm.

The robustness and effectiveness of local representations, when the objects undergo pose change, deformation and partial occlusion [11, 15, 131], have motivated us to propose a structural local 2DDCT sparse appearance model for visual tracking. The proposed algorithm has some similarity to ASLA [15] in the use of local sparse representation, but differs in the domain in which sparse representation is applied. ASLA directly uses the pixel intensities in the local patches for the object appear-

ance model, whereas the proposed appearance model uses the 2DDCT coefficients of those pixel intensities in the local patches. Even though ASLA can handle partial occlusions due to local representations, it does not have any mechanism for occlusion detection, whereas the proposed algorithm has a mechanism to detect occlusion. Also, the proposed algorithm differs from ASLA in terms of the appearance model update. Most of the methods in the literature use pixel intensities in the holistic templates [2, 13] or in the local patches [15, 131, 132] for the object appearance modeling using sparse representation. But the proposed appearance model explores the appearance modeling of the object in the transform domain (2DDCT domain). It is well known that a fraction of the 2DDCT coefficients are sufficient to represent an image with less visual distortion due to the energy compaction property of the 2DDCT [147]. In this section, the energy compaction property of the 2DDCT is exploited by reducing the number of elements/coefficients in the candidate samples and the dictionary to lower the computational cost of the l_1-minimization.

For a given target candidate, the overlapped local patches $\mathbf{P}_i$ inside the target region are extracted with a spatial layout as shown in Figure 4.1. Then, 2DDCT of these patches are computed followed by zigzag scanning, whose order is akin to the one defined in [147]. This gives a matrix $\tilde{\mathbf{Y}} = [\mathbf{y}_{OL}^1, \mathbf{y}_{OL}^2, ..., \mathbf{y}_{OL}^{N_o}] \in \mathbb{R}^{d_o \times N_o}$ for a given candidate, where N_o denotes the number of overlapped local patches extracted within the target region and d_o is the number of pixels in a patch. As each fixed part of the target object is represented by one local patch, the complete holistic structure of a target candidate can be represented by all these N_o local patches with a fixed spatial relationship. Similar procedure is followed for every template in a given set of target templates $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_{n_t}]$ to create a dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{(n_t \times N_o)}] \in \mathbb{R}^{d_o \times (n_t \times N_o)}$, where n_t is the number of target templates. Here, the local patches of the target templates are used as the dictionary atoms to encode the local patches inside the candidate regions, where all these local patches are in 2DDCT domain. As these local patches are obtained across many templates, the resultant dictionary captures

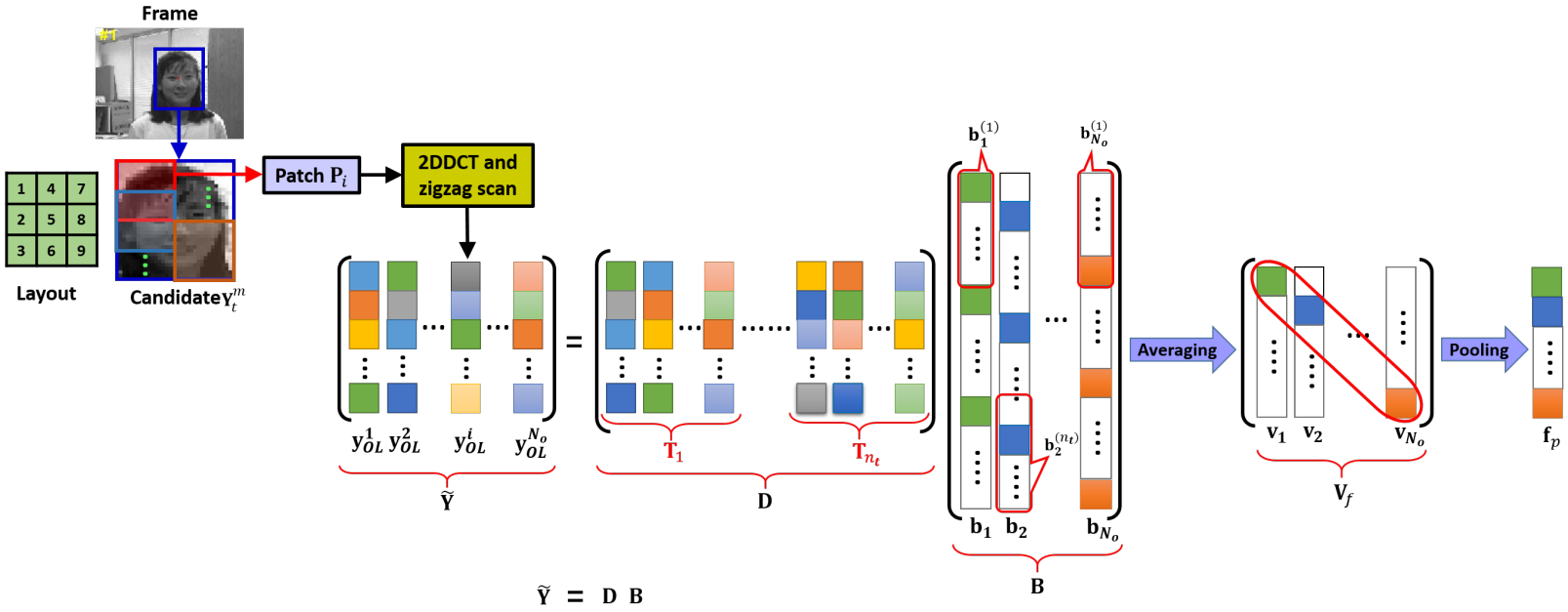


Figure 4.1: Structural local 2DDCT sparse appearance model: illustration of a local patch $\mathbf{P}_i$ extraction followed by 2DDCT, zigzag scanning, averaging of sparse coefficients obtained via the l_1-minimization and alignment pooling of features. 2DDCT of each local patch $\mathbf{y}_{OL}^i$ (where the first patch is denoted in red, second one in blue, and the last one in brown rectangle) is sparsely represented by the patch dictionary $\mathbf{D}$ (in 2DDCT domain) with a sparse vector $\mathbf{b}_i$. These sparse coefficients $\mathbf{b}_i$ are averaged to get feature vectors $\mathbf{v}_i$ and then feature vectors $\mathbf{v}_i$ from all the patches are pooled to represent a target object.

the generality of the different templates and hence, it is able to represent various forms of target parts [15, 134].

In sparse representation, only a few basis elements of the dictionary with different coefficients are sufficient to represent a local patch inside the target region and this is achieved by solving the following minimization problem:

$$\min_{\mathbf{b}_i} \|\mathbf{y}_{OL}^i - \mathbf{D}\mathbf{b}_i\|_2^2 + \lambda\|\mathbf{b}_i\|_1, \quad \text{s.t. } \mathbf{b}_i \succeq 0, \tag{4.1}$$

where $\mathbf{y}_{OL}^i \in \mathbb{R}^{d_o \times 1}$ represents the 2DDCT of the i-th local patch, $\mathbf{b}_i \in \mathbb{R}^{(n_t \times N_o) \times 1}$ is the sparse code of that local patch, and the constraint $\mathbf{b}_i \succeq 0$ indicates that all the elements of $\mathbf{b}_i$ are non-negative. The constraint $\mathbf{b}_i \succeq 0$ is imposed to consider only the positive contributions of the basis elements of the dictionary $\mathbf{D}$ in representing a local patch of the target candidate $\mathbf{y}_{OL}^i$. Now, the sparse codes of the given target candidate is given by $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{N_o}]$, where the sparse coefficients of each local patch $\mathbf{b}_i$ are divided into several segments depending on the template that each element of the vector belongs to, i.e., $\mathbf{b}_i^\mathsf{T} = [\mathbf{b}_i^{(1)\mathsf{T}}, \mathbf{b}_i^{(2)\mathsf{T}}, , \mathbf{b}_i^{(n_t)\mathsf{T}}]$, where $\mathbf{b}_i^{(j)} \in \mathbb{R}^{N_o \times 1}$ indicates the j-th segment of the sparse coefficient vector $\mathbf{b}_i$ corresponding to the template $\mathbf{T}_j$ in the given target template set $\mathbf{T}$ (as shown in Figure 4.1). From these segmented sparse coefficients $\mathbf{b}_i^{(j)}$, a normalized feature vector $\mathbf{v}_i \in \mathbb{R}^{N_o \times 1}$ for the i-th local patch is obtained as

$$\mathbf{v}_i = \frac{1}{G} \sum_{j=1}^{n_t} \mathbf{b}_i^{(j)}, \quad i = 1, 2, ..., N_o, \tag{4.2}$$

where G is a normalization term, which makes all the contributions from the templates sum to unity. Thus, for a given candidate, all the normalized feature vectors of the local patches within a candidate region form a square matrix $\mathbf{V}_f = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{N_o}] \in \mathbb{R}^{N_o \times N_o}$. Since a single local patch captures only some local appearance of the object, the whole object modeling requires pooling of information from these normalized feature vectors. Here, alignment pooling is chosen due to its capability of using full structural information contained in the dictionary and precisely locating the target

object [15]. Even though each local patch at a given position of the candidate is represented by patches at different positions of the templates, the local appearance of a patch with some appearance variation in a candidate is correctly represented by the patches at the same positions of the templates, i.e, the top left corner patch of the object in Figure 4.1 can be represented precisely by the top left corner patches of the templates. This is achieved by considering only the diagonal elements of the square matrix $\mathbf{V}_f$ as the pooled feature vector $\mathbf{f}_p \in \mathbb{R}^{N_o \times 1}$, given by

$$\mathbf{f}_p = \text{diag}(\mathbf{V}_f) \tag{4.3}$$

This feature vector $\mathbf{f}_p$ not only captures the target structure with a fixed spatial relationship but also reflects the similarity between the candidate and the target template.

4.3 Holistic Image Reconstruction from an Overlapped Local Patches

In the proposed appearance model, the overlapped local patches are used for the object representation rather than holistic templates due to their robustness to pose change, deformation and occlusion. The overlapped local patches of size 16×16 with an overlap of 8 pixels are extracted from an image of size 32×32 as per the layout shown in Figure 4.1. Now, the extracted local patches are concatenated as per their spatial relationship to obtain a overall block of size 48×48 (as shown in left part of Figure 4.2). For clarity, the original image of size 32×32 is assumed to be divided into a sub-blocks of size 8×8 as shown in right part of Figure 4.2. Now, the first 16×16 patch extracted from the top-left corner of the image, denoted as block **1** in left part of Figure 4.2, is comprised of four 8×8 sub-blocks denoted as sub-block *A, B, E* and *F* in right part of Figure 4.2. Similarly, the second block, denoted as block

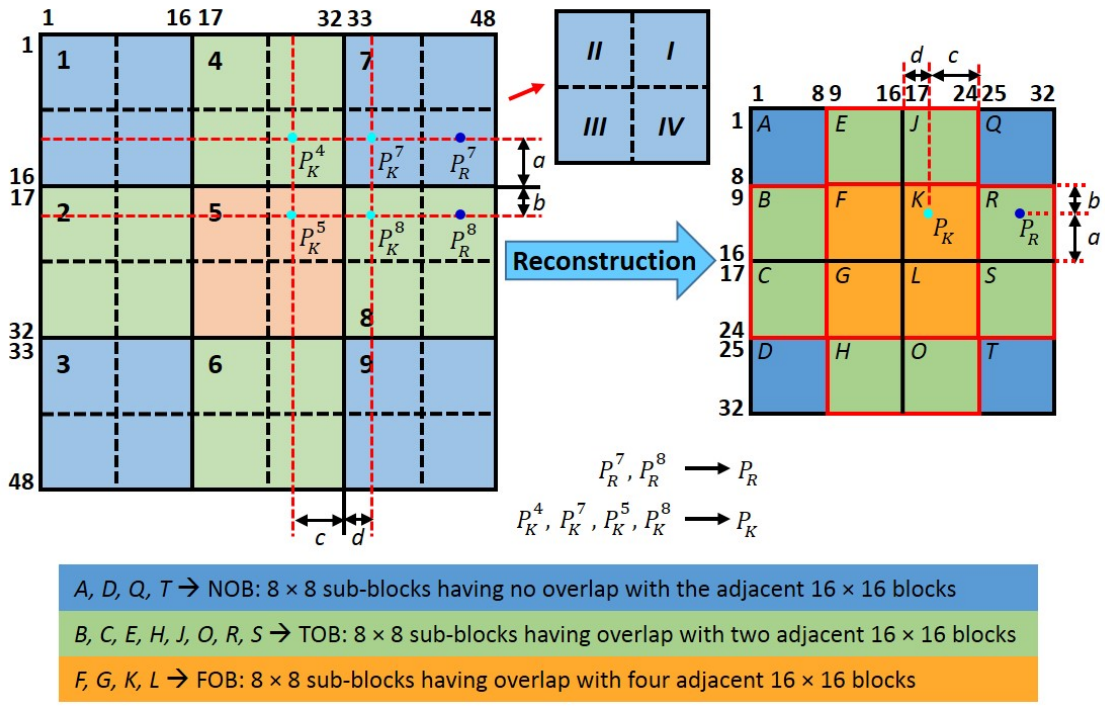


Figure 4.2: Holistic image reconstruction from overlapped local patches.

2, is comprised of four 8×8 sub-blocks *B, C, F* and *G*, and so on.

To understand the reconstruction procedure, the 8×8 sub-blocks in Figure 4.2 are divided into three groups depending on the number of overlaps they have with the neighboring 16×16 blocks. The first group consisting of four 8×8 sub-blocks (*A, D, Q* and *T*) has no overlaps with 16×16 blocks, and are denoted as no overlapping blocks (NOB) (blue color blocks in the right part of Figure 4.2). The second group consisting of eight 8×8 sub-blocks (*B, C, E, H, J, O, R* and *S*) has an overlap with two adjacent 16×16 blocks, and are denoted as two overlapping blocks (TOB) (green color blocks in the reconstructed image of Figure 4.2). Similarly, the last group consisting of four 8×8 sub-blocks (*F, G, K* and *L*) has an overlap with four adjacent 16×16 blocks, and are denoted as four overlapping blocks (FOB) (orange color blocks in the reconstructed image of Figure 4.2). Since the NOB group has no overlaps, the pixels in the 8×8 sub-blocks *A, D, Q* and *T* of the reconstructed image are copied directly

from the *II*-quadrant of block **1**, *III*-quadrant of block **3**, *I*-quadrant of block **7** and *IV*-quadrant of block **9**, respectively. However, the TOB group has two overlapping 16×16 blocks and hence the pixels in the 8×8 sub-blocks *B, C, E, H, J, Q, R* and *S* of the reconstructed image are computed from the corresponding two overlapping 16×16 blocks. For example, the 8×8 sub-block *R* is reconstructed from *IV*-quadrant of block **7** and *I*-quadrant of block **8**. Therefore, all the pixels in the *IV*-quadrant of block **7** (denoted as P_R^7) and *I*-quadrant of block **8** (denoted as P_R^8) are used to find the pixel values in 8×8 sub-block *R* (denoted as P_R) by

$$P_R = \frac{a\, P_R^7 + b\, P_R^8}{a + b}, \tag{4.4}$$

where a and b are the distances as shown in Figure 4.2. Similarly, all the pixels in the 8×8 sub-blocks *F, G, K* and *L* belonging to FOB group are computed from the pixels of the corresponding four overlapping 16×16 blocks using

$$P_K = \frac{c}{c + d}\left(\frac{a\, P_K^4 + b\, P_K^5}{a + b}\right) + \frac{d}{c + d}\left(\frac{a\, P_K^7 + b\, P_K^8}{a + b}\right), \tag{4.5}$$

where a, b, c and d are the distances as shown in Figure 4.2, P_K^4, P_K^5, P_K^7 and P_K^8 are the pixel values from the blocks **4, 5, 7** and **8**, respectively, and P_K is the reconstructed pixel belonging to block *K*. In this way, all the 8×8 sub-blocks are obtained to get the reconstructed holistic image $\mathbf{Y}_r$. Figure 4.3 shows the reconstructed holistic image $\mathbf{Y}_r$ of the *Jogging-2* and *Woman* sequences to illustrate the effectiveness of the proposed image reconstruction procedure without introducing any errors/artifacts during transition from one patch to another.

4.4 Proposed Tracking Algorithm

The 2DDCT of the overlapped local patches inside the target region are used to model the object appearance, and these overlapped local patches can be represented by a

very few low-frequency 2DDCT coefficients, which can preserve the image information in a patch very well due to the energy compaction property of the 2DDCT. Hence, only r_o lower frequency 2DDCT coefficients are considered out of d_o coefficients in all the patches of the candidates and the dictionary while solving the l_1-minimization problem in (4.1). The confidence scores C^m for all the candidates are computed from the sparse codes using (4.2), (4.3) as

$$C^m = \sum_{i=1}^{N_o} \mathbf{f}_p^m(i) \tag{4.6}$$

In visual tracking based on particle filter framework, the confidence of the each particle is given by its observation likelihood, defined as

$$p(\mathbf{Y}_t^m | \mathbf{s}_t^m) \propto C^m \tag{4.7}$$

Finally, the optimal state of the target $\hat{\mathbf{s}}_t$ is estimated using (2.4). Further, the observation models are adapted to handle the appearance change of the target by incrementally updating the template set and dictionary, as discussed in the next subsection.

4.4.1 Observation Model Update

The update of observation model is very much essential to handle the appearance variations of the object, but the update with imprecise samples will cause tracking drift due to model degradation. Therefore, the imprecise samples should be avoided during the model update. Even though ASLA is efficient during partial occlusion due to its local appearance model, its template update mechanism has a drawback during occlusion. That is, the tracking results, which are occluded, are employed directly for an incremental subspace learning thereby degenerating the PCA subspace. As the image reconstructed from the degenerated PCA subspace is used for the template

update, there are chances of appearance model degradation (see Figure 4.4a) resulting in a tracking drift. To address this issue, it is proposed to detect and generate a robust occlusion map, and use it to modify the occluded samples before the observation model update.

For the occlusion map generation, all the 2DDCT coefficients are used in both the dictionary $\mathbf{D}$ and the candidate matrix $\tilde{\mathbf{Y}}$ of the tracked target candidate image $\hat{\mathbf{Y}}$ to compute the sparse codes $\hat{\mathbf{B}}$ using (4.1). The local appearance of a patch in a target candidate is correctly represented by the patches at the same positions of the templates. Hence, the sparse codes corresponding to the respective patches are only used to compute the overlapped local patches. These overlapped patches are used to reconstruct the holistic image $\mathbf{Y}_r$ as described in section 4.3 and then the holistic reconstruction error $\mathbf{E}_h = \hat{\mathbf{Y}} - \mathbf{Y}_r$ is computed. Further, the holistic reconstruction error $\mathbf{E}_h$ is used to generate a binary occlusion map $\mathbf{O}_1$ using (4.8) indicating one for occluded pixels and zero for non-occluded pixels.

$$\mathbf{O}_1 = \begin{cases} 1, & \text{if } |\mathbf{E}_h| \geq O_{thr} \\ 0, & \text{otherwise,} \end{cases} \tag{4.8}$$

where O_{thr} is a precomputed threshold that decides whether the pixel is occluded or not. In order to compute O_{thr}, it is assumed that all the elements of the reconstruction error $\mathbf{E}_h$ will be in the "normal range" and follow the Gaussian distribution in the absence of occlusion. But during occlusions, the occluded elements of the reconstruction error $\mathbf{E}_h$ will probably exceed the "normal range". Therefore, by knowing the "normal range" of the reconstruction error in the initial frames (from 1 to n_t) of the respective sequence, the value of the O_{thr} is computed as

$$O_{thr} = \frac{c_3}{n_t} \sum_{f=1}^{n_t} \mathsf{std}\ (\mathbf{E}_h(f)) \tag{4.9}$$

where c_3 is a constant and std represents the standard deviation. Here, each target

template in a template set $\mathbf{T}$ is used as the candidate sample and the remaining $n_t - 1$ templates are used as the dictionary in a round-robin fashion to compute O_{thr}. In general, the occlusion is a large connected region as opposed to the random noises or object appearance variations, whose region is very small. Hence, the occlusion map is updated to retain only the large connected region by applying a morphological operations and a connected component analysis, to fill the small hole between the regions and remove the small regions.

In order to increase the robustness of the occlusion detection method, it is proposed to identify the patch of the tracked target candidate with no contribution from the respective patches of the dictionary $\mathbf{D}$ using the pooled feature vector $\hat{\mathbf{f}}_p$, which is obtained from sparse codes $\hat{\mathbf{B}}$ using (4.2) and (4.3). Note that if there is no contribution from the respective patches of the dictionary $\mathbf{D}$, the respective element of the pooled feature vector $\hat{\mathbf{f}}_p$ will be zero. That is, $\hat{\mathbf{f}}_p(i) = 0$ indicates that there is no contribution from the i-th patch of all the templates in representing the corresponding patch of the tracked target candidate, and this happens when the i-th patch is occluded fully. Then, a binary occlusion map $\mathbf{O}_2$ is generated by indicating one in the respective pixel locations of the corresponding patch.

Now, the two occlusion maps $\mathbf{O}_1$ and $\mathbf{O}_2$ are combined to generate a final occlusion map $\mathbf{O}$ by performing a logical-OR operation. This makes the occlusion map more robust, so that the chances of appearance model deterioration due to occlusion could be reduced. Figure 4.3 shows the occlusion maps $\mathbf{O}_1$, $\mathbf{O}_2$ and $\mathbf{O}$ for *Jogging-2* (#45) and *Woman* (#122) sequences. Further, a occlusion ratio τ is computed as the ratio of the number of ones in $\mathbf{O}$ to the total number of elements in $\mathbf{O}$. This occlusion ratio τ indicates the amount of occlusion in the tracked candidate. The occlusion ratio τ decides whether the update of the observation model with the tracked sample is full or partial or not, with the help of two thresholds τ_1 and τ_2. In the absence of occlusion (when $\tau < \tau_1$), the tracked sample is used directly for the model update (full update). During partial occlusion (when $\tau_1 < \tau < \tau_2$), the occluded pixels in

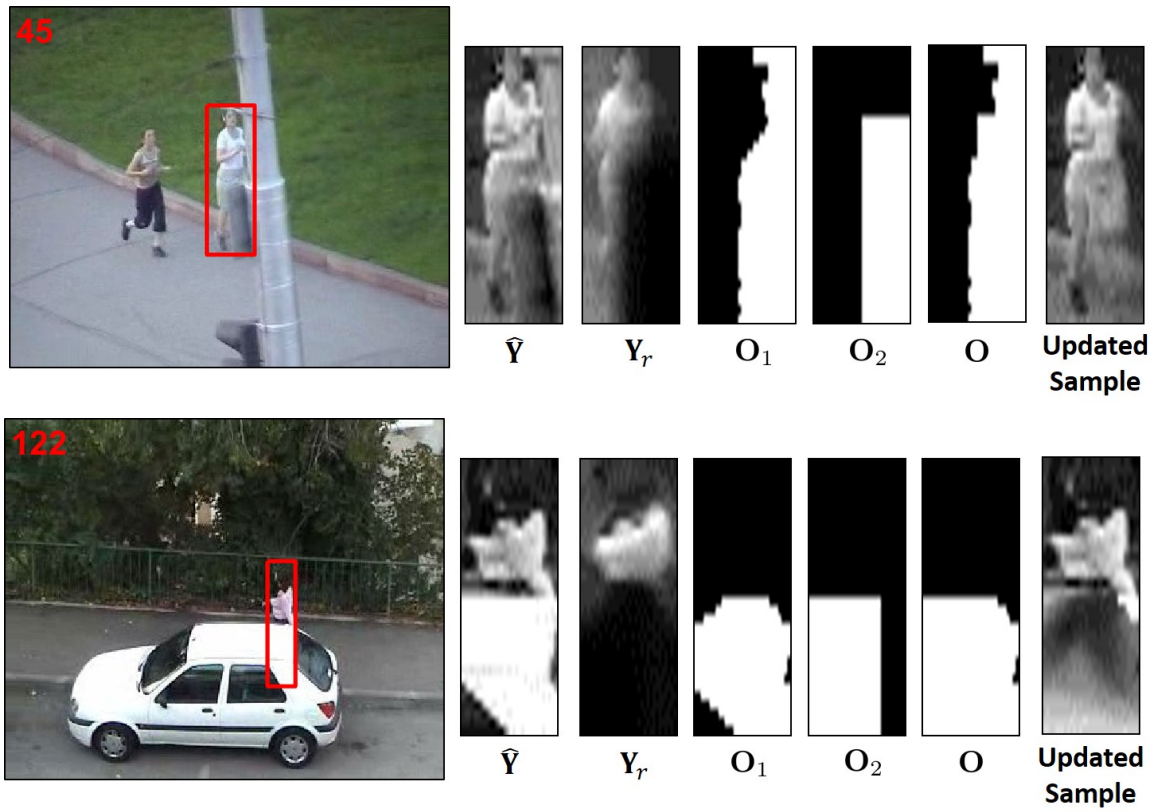


Figure 4.3: Some representative cases of *Jogging-2* (#45) and *Woman* (#122) sequences to illustrate the effectiveness of the proposed holistic image reconstruction, the robust occlusion map generation ($\mathbf{O}_1$, $\mathbf{O}_2$, $\mathbf{O}$), and the updated sample.

the tracked sample are replaced with the corresponding pixels from the previously updated PCA mean vector $\boldsymbol{\mu}_{LD}$ to get a updated sample. This updated sample is free from occlusion and is used in the model update. During severe occlusion (when $\tau > \tau_2$), the tracked sample is not used for the model update. Figure 4.3 shows the updated tracked sample, which is free from occlusion, for *Jogging-2* (#45) and *Woman* (#122) sequences. It is observed from Figure 4.3 that the combined occlusion map $\mathbf{O}$ is more robust than the individual ones $\mathbf{O}_1$ and $\mathbf{O}_2$, and make the updated sample free from occlusion.

The updated tracked samples, which are free from occlusion, are cumulated and then used to update the PCA subspace model ($\boldsymbol{\mu}_{LD}$, $\mathbf{U}_{LD}$) by an incremental subspace

(a) Template set of ASLA

(b) Template set of the proposed algorithm

Figure 4.4: Template set for some representative cases of *Faceocc1* (#700) and *Woman* (#180) to illustrate the effectiveness of the proposed observation model update compared to that of ASLA.

learning [1]. This incremental learning not only adapts to the target appearance variation but also preserves the common visual information in the collected observations. As in ASLA [15], the proposed observation model update uses the PCA mean vector $\boldsymbol{\mu}_{LD}$ and the trivial templates along with PCA basis vectors $\mathbf{U}_{LD}$ to estimate the target $\mathbf{p}$, as given by

$$\mathbf{p} = \boldsymbol{\mu}_{LD} + \mathbf{U}_{LD}\,\mathbf{q} + \mathbf{e} = \boldsymbol{\mu}_{LD} + [\ \mathbf{U}_{LD} \quad \mathbf{I}\ ] \begin{bmatrix} \mathbf{q} \\ \mathbf{e} \end{bmatrix} \tag{4.10}$$

where $\mathbf{I}$ is the identity matrix representing trivial templates, $\mathbf{q}$ denotes the coefficients of the PCA basis vectors $\mathbf{U}_{LD}$, and $\mathbf{e}$ represents the pixels in $\mathbf{p}$ that are outliers or corrupted. Unlike ASLA [15], which uses the latest tracked candidate as $\mathbf{p}$, the proposed observation model update choses the updated tracked sample with highest confidence score among the cumulated updated tracked samples as $\mathbf{p}$, which is free from occlusion. Note that when $\tau > \tau_2$, the tracked candidate with severe occlusion

is not cumulated and hence its probability of considering as $\mathbf{p}$ is zero. Since the error is arbitrary and sparse, (4.10) is solved by l_1-minimization, and an image $\hat{\mathbf{p}}$ is reconstructed from the PCA mean vector $\boldsymbol{\mu}_{LD}$ and the PCA basis vectors $\mathbf{U}_{LD}$ along with its coefficients $\mathbf{q}$. This ensures the reconstructed image $\hat{\mathbf{p}}$ is free from corruption and outliers, as the coefficients of trivial templates $\mathbf{e}$ (due to noise or outlier) are excluded for image reconstruction. The reconstructed image $\hat{\mathbf{p}}$ is then used for updating both the template and the corresponding dictionary atoms in $\mathbf{D}$ as discussed in the next paragraph.

It is known that the sparse codes $\hat{\mathbf{B}}$ of the tracked target candidate represent the contributions from all the patches of all the templates. Also, the local appearance of a patch with some appearance variation in the tracked target candidate is correctly represented by the patches at the same positions of the templates. Hence, the contribution $\hat{\mathbf{h}}_j$ of the $j-$th template in representing the tracked target candidate $\hat{\mathbf{Y}}$ is computed by considering only the contributions from the respective patches of the $j-$th template using (4.11).

$$\hat{\mathbf{h}}_j = \sum_{i=1}^{N_o} \hat{\mathbf{B}}\left(N_o[j-1]+i, i\right), \quad j = 1, 2, ..., n_t, \tag{4.11}$$

While cumulating the updated tracked samples, the respective confidence $\hat{C}$ and the template contribution scores $\hat{\mathbf{h}}_j$ are also cumulated. From the cumulated confidence scores, the highest confidence score is found and its corresponding updated tracked sample is used as $\mathbf{p}$ in (4.10). Further, from the cumulated template contribution scores, the location $\hat{j}$ of the template to be replaced is found using the (4.12).

$$\hat{j} = \arg\min_{j} \sum_{t \in [t-4:t]} \hat{\mathbf{h}}_j(t) \tag{4.12}$$

With this replacement strategy, the template, which is contributing least to the representation of the previous 5 tracking results, is replaced with the reconstructed image

$\hat{\mathbf{p}}$. This is done with an assumption that the template with the least contribution score may have an old appearance of the object, which may be outdated, and hence cannot contribute significantly in representing the target candidate. After updating the template set with $\hat{\mathbf{p}}$, the corresponding dictionary atoms in $\mathbf{D}$ are also updated. To illustrate the effectiveness of the proposed observation model update, the template set for the proposed algorithm and ASLA is shown in Figure 4.4. From Figure 4.4, it is observed that the template set of ASLA gets corrupted over the time in contrast to that of the proposed algorithm. Unlike [2], which considers only the current tracking result (may be severely occluded) to find the template contribution score, the proposed algorithm considers previous 5 tracking results (which are not severely occluded with $\tau < \tau_2$) to find the average template contribution score. As [2] directly uses the occluded/corrupted tracking result for the model update, there are chances of observation model degradation. However, in the proposed algorithm, $\hat{\mathbf{p}}$ used for the model update is free from the occlusion and outliers/corruptions, as they are removed before and after an incremental subspace learning. Further, in [43] the occlusion is detected based on the patch reconstruction error as opposed to that based on holistic reconstruction error in the proposed algorithm.

In the proposed algorithm, instead of using the sum of pooled features to find the confidence score C^m by giving equal weights to each patch (as in ASLA), weighted sum of the pooled features are used to compute the confidence score C^m. This is done by assigning different weights to each patch depending on its patch occlusion ratio τ_{p_i}. Now the confidence score C^m is rewritten as

$$C^m = \sum_{i=1}^{N_o} \tau_{p_i} \mathbf{f}_p^m(i) \tag{4.13}$$

where τ_{p_i} is the ratio of occluded pixels to that of total pixels in a patch. The proposed tracking method based on the structural local 2DDCT sparse appearance model and the occlusion handling mechanism is summarized in Algorithm 4.1.

Algorithm 4.1 Tracking algorithm based on the structural local 2DDCT sparse appearance model and the occlusion handling mechanism

Input: Target object is labeled in the first frame and its initial state is $\mathbf{s}_1$, number of templates n_t, number of local patches N_o.

1: Collect a set of n_t templates, $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_{n_t}]$, using kd-tree to model the object appearance.
2: From every template in $\mathbf{T}$, extract the N_o overlapped local patches $\mathbf{P}_i$, compute the 2DDCT followed by zigzag scanning to create a dictionary $\mathbf{D}$.
3: **for** $t > n_t$ **do**
4: Sample M candidate states $\{\mathbf{s}_t^1, \mathbf{s}_t^2, ..., \mathbf{s}_t^M\}$ from $\mathbf{s}_{t-1}$.
5: For every candidate state $\mathbf{s}_t^m$, extract the corresponding image sample $\mathbf{Y}_t^m$ and candidate matrix in 2DDCT domain $\tilde{\mathbf{Y}}_t^m$ as explained in section 4.2.
6: For all the candidate samples $\mathbf{Y}_t^m$, compute sparse codes $\mathbf{b}_i$ for each local patch vector $\mathbf{y}_{OL}^i$ in the candidate sample matrix $\tilde{\mathbf{Y}}_t^m$ using (4.1) by considering only r_o number of 2DDCT coefficients in both candidates and the dictionary.
7: Compute the pooled feature vector $\mathbf{f}_p^m$ and the confidence score C^m for all the candidates using (4.2), (4.3) and (4.13), respectively.
8: Find the optimal state of the tracked target $\hat{\mathbf{s}}_t$ using the (4.7) and (2.4).
9: Update the template set $\mathbf{T}$ and dictionary $\mathbf{D}$ as described in section 4.4.1 for every 5 frames.
10: **end for**

Output: Target state $\hat{\mathbf{s}}_t$ at time t.

4.5 Experimental Results

Algorithm 4.1 is implemented in MATLAB and its performance is evaluated on the challenging sequences available in OTB-50 [7] and VOT2016 [94] datasets using the respective evaluation protocols discussed in Section 2.3. In Algorithm 4.1, each image observation is resized to 32×32 pixels and then local patches of size 16×16 are extracted with an overlap of 8 pixels. Therefore, each target region is cut into $N_o = 9$ overlapping patches. In Algorithm 4.1, sparse modeling software (SPAMS) package [148] is used for l_1-norm minimization and the regularization constant λ is set to 0.01. 10 eigenvectors are used in an incremental subspace learning and the observation model is updated for every 5 frames. Considering the trade-off between effectiveness in tracking and computational efficiency, $M = 600$ particles are sampled using a particle filter, which is discussed in Section 2.2. The constant c_3 used to compute threshold O_{thr} in (4.9) is set to 4. The occlusion ratio thresholds τ_1 and τ_2 are set to 0.1 and 0.65, respectively. In the initial 10 frames, kd-tree is used to obtain the tracking results, and from these tracking results $n_t = 10$ target templates are extracted for the generation of the dictionary $\mathbf{D}$. The number of 2DDCT coefficients considered in each patch, r_o, is set to 64 while computing confidence score C^m for all the candidates, except during occlusions when $\tau > \tau_2$, it is set to 256. By reducing the size of the dictionary as well as that of the candidate samples, Algorithm 4.1 has achieved a speed of 2.18 fps[1] (including the time required for the computation of the 2DDCT and the proposed occlusion detection) as compared with 1.86 fps required by ASLA [15]. This is a 17.2% increase in the speed compared to that of ASLA in spite of having to compute the 2DDCT coefficients and to detect the occlusion.

The performance of Algorithm 4.1 is evaluated against several state-of-the-art tracking algorithms, ASLA [15], local 2DDCT sparse appearance model (LDSAM) [134], IL3DDCT [36] and the top two best tracking algorithms selected from the

[1]Using modern computer of 3.4GHz CPU and 16GB RAM

previous chapter based on their performance on each of the challenging attributes of both the OTB-50 and VOT2016 benchmark datasets, for a fair comparison. The codes of the trackers ASLA [15] and IL3DDCT [36] downloaded from the respective authors' website [149] and [150] are used to evaluate on the sequences of the benchmarks to have a fair comparison with Algorithm 4.1. The parameter settings of these trackers are as given in their respective papers in all the experiments, except for the parameters related to the particle filter framework, which are set to be as in OTB-50 [7] for a fair comparison. In the following two sub-sections, two benchmark datasets and the evaluation measures discussed in Section 2.3 are used for the quantitative evaluation of Algorithm 4.1 with that of the other methods.

4.5.1 Quantitative Evaluation on OTB-50

The performance of Algorithm 4.1 is evaluated on the OTB-50 benchmark [7] and compared with the state-of-the-art tracking algorithms using one-pass evaluation (OPE). Table 4.1 shows the performance comparison of Algorithm 4.1 in terms of the *precision score* for the threshold of 20 pixels with that of the considered trackers for different attributes. The best three results are shown in **(red, bold)**, (violet, underline) and *(blue, italic)* fonts for better comparison of Algorithm 4.1 with the other trackers. It is observed that Algorithm 4.1 ranks first in illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV) and occlusion (Occ) challenging attributes, second in deformation (Def), in-plane rotation (IPR) and low resolution (LR), and third in background clutter (BC) challenging attributes. On the other hand, LDSAM [134], stands first in Def and IPR challenging attributes, stood second in IV, OPR, Occ, out-of-view (OV) and BC challenging attributes. Also, Algorithm 3.1_HOG ranks first in OV and third in IV, SV, motion blur (MB), fast motion (FM) and IPR challenges. Further, ASLA ranks first in BC, second in SV and third OPR and Def challenging attributes. Also, DLT stands first in MB and FM, third in Occ and OV attributes, and LSST stands first in LR attributes. L1PAG

ranks second in MB and FM, and third in LR challenging attributes. It is also observed from the Table 4.1 that the performance of Algorithm 4.1 is superior to that of ASLA in the face of all the challenges except OV and BC. Algorithm 4.1, is an improvement of LDSAM [134] with a robust occlusion detection mechanism, which is used to update the observation model as well as to find the confidence score. This has resulted in an improved performance of Algorithm 4.1 over that of LDSAM in IV, OPR, SV, Occ, MB, FM and LR challenging attributes. It is also observed that Algorithm 4.1 has a performance better than that of Algorithm 3.1_HOG for all the challenging attributes except MB, FM and OV. Further, Algorithm 4.1 outperforms all the methods for most of the challenging attributes, except Def, MB, FM, IPR, OV, BC and LR.

The performance comparison of Algorithm 4.1 in terms of area under curve (AUC) with that of the considered trackers for the different attribute challenges is given in Table 4.2. It is noticed that Algorithm 4.1 stands first in IV, OPR, SV, Occ and LR challenging attributes, stood second in Def and third in BC challenging attributes. LDSAM ranks first in Def, second in IV, OPR, Occ, IPR, OV and BC, and third in LR attributes. Also, Algorithm 3.1_HOG stands first in IPR and OV, stood second in MB, and third in IV, SV, Occ, Def and FM attributes. Further, ASLA stands first in BC, stood second in SV and Def, and third in IV, OPR, IPR and OV attributes. Also, DLT stands first in MB and FM attributes, LSST stood second in LR, whereas L1APG stood second in FM and third in MB. Algorithm 4.1 has a performance better than that of ASLA in face of all the challenges except FM, IPR, OV and BC. Due to the robust occlusion detection mechanism, the performance of Algorithm 4.1 in terms of AUC has been improved to that of LDSAM [134] in IV, OPR, SV, Occ, MB, FM and LR challenging attributes. It is also noticed that Algorithm 4.1 has a performance superior to that of Algorithm 3.1_HOG for all the challenging attributes except MB, FM, IPR and OV. Further, Algorithm 4.1 outperforms all the methods for most of the challenging attributes, except Def, MB, FM, IPR, OV and BC. This

Table 4.1: The *precision score* of Algorithm 4.1 with that of the compared trackers for different attributes of OTB-50. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Precision score	**IV**	**OPR**	**SV**	**Occ**	**Def**	**MB**	**FM**	**IPR**	**OV**	**BC**	**LR**
Algorithm 4.1	**61.5**	**62.9**	**63.4**	**60.4**	59.8	33.9	33.3	58.6	39.9	*60.1*	68.1
LDSAM	58.8	61.6	60.1	58.1	**59.9**	29.9	28.4	**59.4**	45.9	62.4	61.2
Algorithm 3.1_HOG	*56.5*	56.3	*60.3*	55.8	52.6	*36.5*	*35.1*	*58.3*	**51.0**	57.4	55.7
Algorithm 3.1_Gray	42.8	50.4	53.2	52.3	49.5	26.2	24.8	48.5	36.3	44.8	60.2
ASLA	56.4	*59.1*	62.3	53.3	*57.7*	32.3	33.1	57.5	41.7	**63.6**	59.0
IL3DDCT	52.7	51.4	51.3	45.6	49.0	27.2	24.6	51.0	26.1	46.7	59.9
L1APG	34.1	47.8	47.2	46.1	38.3	37.5	36.5	51.8	32.9	42.5	*61.5*
WRMPCA	42.7	47.7	52.8	45.9	41.2	24.3	23.6	47.5	31.1	43.4	58.7
LWIST	34.3	38.4	40.3	40.4	35.3	15.1	18.7	33.7	30.7	37.4	45.9
WLCS	32.9	33.0	38.3	35.7	29.9	15.4	21.3	30.3	22.8	33.1	23.0
LSST	41.5	47.0	53.3	41.3	45.0	27.1	25.2	46.1	24.3	42.6	**74.1**
DLT	53.4	56.1	59.0	*57.4*	56.3	**45.3**	**44.6**	54.8	*44.4*	49.5	53.6
PCOM	36.8	46.1	48.6	44.4	38.3	22.9	22.0	46.0	30.5	40.1	55.9

may happen when the target undergoes a large appearance change due to either fast motion or out-of-view or in-plane-rotation of the object, and when there is a reduction in information of the object due to motion blur.

The performance comparison of Algorithm 4.1 using the *precision* and *success plots* of OPE for OTB-50 benchmark sequences having occlusion against the other trackers is shown in Figure 4.5. In the *precision plot*, the *precision score* for the location error threshold of 20 pixels is used to rank the tracker, whereas in the *success plot*, AUC is used to rank the overall performance of the tracker. From Figure 4.5, it is observed that Algorithm 4.1 outperforms the other trackers LDSAM, DLT, Algorithm 3.1_HOG, ASLA, Algorithm 3.1_Gray, L1APG, WRMPCA, IL3DDCT, PCOM and LSST by 3.9%, 5.2%, 8.2%, 12.9%, 15.5%, 23.6%, 31.6%, 32.4%, 36% and 46.2%, respectively, and LDSAM [134] performs better than DLT, Algorithm 3.1_HOG, ASLA, Algorithm 3.1_Gray, L1APG, WRMPCA, IL3DDCT, PCOM and LSST do by 1.2%, 4.1%, 9.0%, 11.0%, 26.0%, 26.3%, 27.4%, 30.8% and 40.6%, respectively, in terms of the *precision score*. Similarly, Algorithm 4.1 outperforms the other trackers LD-

Table 4.2: AUC of Algorithm 4.1 with that of the compared trackers for different attributes of OTB-50. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

AUC	IV	OPR	SV	Occ	Def	MB	FM	IPR	OV	BC	LR
Algorithm 4.1	**49.3**	**47.8**	**49.4**	**46.6**	46.2	28.7	28.6	45.1	35.2	*47.1*	**47.2**
LDSAM	47.9	47.5	47.5	45.9	**48.0**	27.3	26.5	46.3	40.4	49.4	*44.1*
Algorithm 3.1_HOG	*46.6*	44.6	*48.2*	*44.0*	*42.0*	31.6	*30.2*	**46.4**	**43.2**	46.4	41.6
Algorithm 3.1_Gray	33.7	37.0	38.6	37.7	34.6	23.3	22.3	37.1	32.3	31.3	41.7
ASLA	*46.6*	*46.5*	49.0	43.1	46.2	28.2	29.2	*45.6*	*38.7*	**50.2**	42.1
IL3DDCT	37.2	35.8	33.9	34.3	34.3	24.6	22.3	36.8	27.6	32.5	34.1
L1APG	28.3	36.0	35.0	35.3	31.1	*31.0*	31.1	39.1	30.3	35.0	37.4
WRMPCA	33.2	34.7	37.4	33.8	29.7	20.8	21.6	35.1	27.4	31.6	38.9
LWIST	28.9	30.1	31.3	31.9	28.5	15.1	18.3	27.2	28.1	31.1	27.8
WLCS	27.0	25.8	29.5	27.6	25.0	15.8	19.6	24.2	21.6	27.3	13.7
LSST	33.2	33.5	37.7	30.4	31.6	21.4	21.5	32.3	21.7	30.2	46.2
DLT	40.5	41.2	45.5	42.3	39.4	**36.3**	**36.0**	41.1	36.7	33.9	34.7
PCOM	28.4	32.3	32.9	33.2	26.0	17.8	18.8	32.4	27.5	28.4	34.5

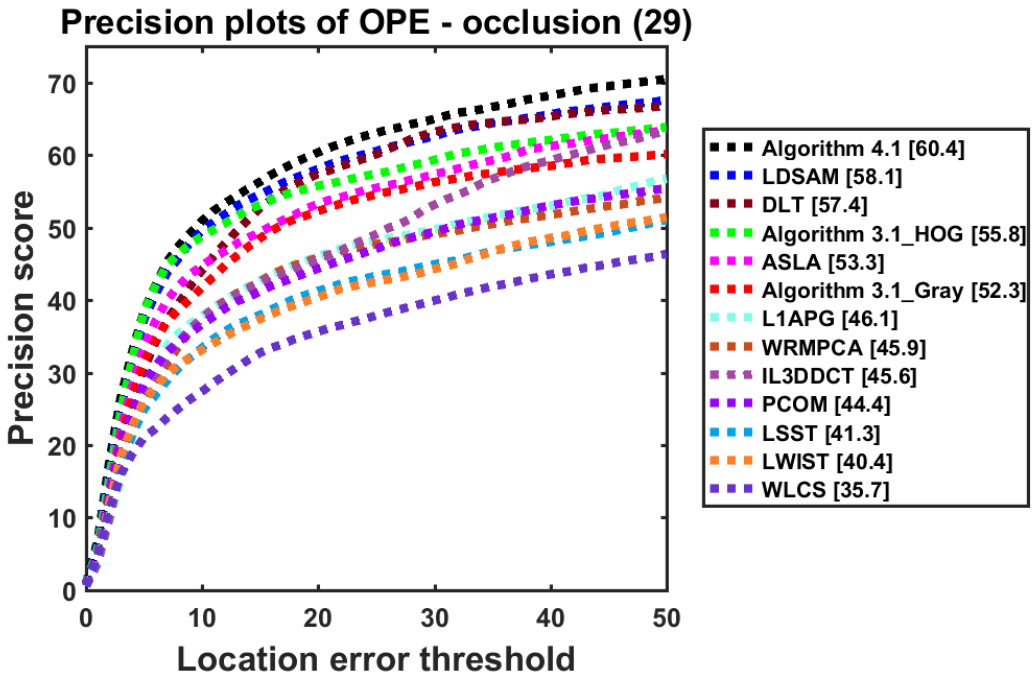
SAM, Algorithm 3.1_HOG, ASLA, DLT, Algorithm 3.1_Gray, L1APG, IL3DDCT, WRMPCA, PCOM and LWIST by 1.5%, 5.9%, 8.1%, 10.1%, 23.6%, 32.0%, 35.8%, 37.8%, 40.3% and 46.0%, respectively, and the performance of LDSAM is superior to that of Algorithm 3.1_HOG, ASLA, DLT, Algorithm 3.1_Gray, L1APG, IL3DDCT, WRMPCA, PCOM and LWIST by 4.3%, 6.5%, 8.5%, 21.7%, 30.0%, 33.8%, 35.7%, 38.2% and 43.8%, respectively, in terms of AUC. For sequences having occlusion, it is observed that Algorithm 4.1 and LDSAM [134] have shown a performance better than that of the other trackers and Algorithm 3.1 in terms of both the *precision score* and AUC.

The *precision* and *success plots* of OPE for the various trackers averaging over the OTB-50 benchmark sequences are shown in Figure 4.6. From Figure 4.6, it is observed that Algorithm 4.1 outperforms the state-of-the-trackers LDSAM, ASLA, Algorithm 3.1_HOG, DLT, Algorithm 3.1_Gray, IL3DDCT, WRMPCA, LSST, L1APG and PCOM by 0.9%, 3.7%, 3.9%, 8.0%, 15.2%, 23.6%, 23.6%, 28.3%, 30.7% and 33.2%, respectively, and LDSAM performs better than ASLA, Algorithm 3.1_HOG,

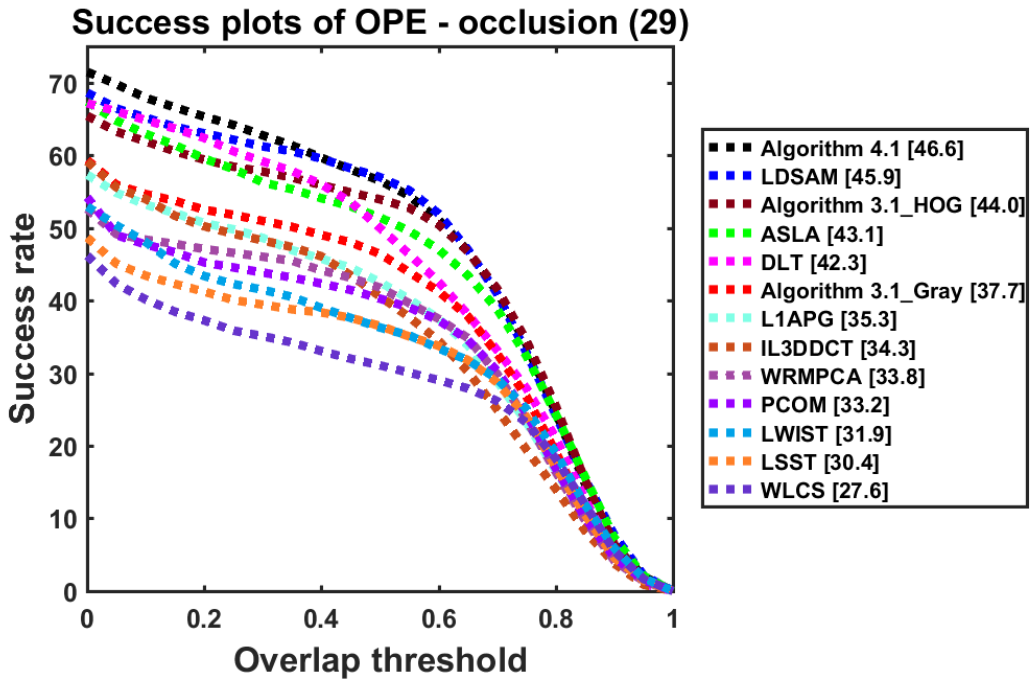
DLT, Algorithm 3.1_Gray, IL3DDCT, WRMPCA, LSST, L1APG and PCOM do by 2.7%, 2.9%, 6.9%, 14.1%, 22.4%, 22.4%, 27.1%, 29.4% and 31.9%, respectively, in terms of the *precision score*. Similarly, the performance of Algorithm 4.1 is superior to that of ASLA, Algorithm 3.1_HOG, DLT, Algorithm 3.1_Gray, WRMPCA, L1APG, IL3DDCT, LSST and PCOM by 1.2%, 1.8%, 13.3%, 21.9%, 29.6%, 30.0%, 36.4%, 38.7% and 45.3%, respectively, and LDSAM [134] outperforms ASLA, Algorithm 3.1_HOG, DLT, Algorithm 3.1_Gray, WRMPCA, L1APG, IL3DDCT, LSST and PCOM by 1.4%, 2.0%, 13.5%, 22.2%, 29.9%, 30.2%, 36.7%, 39.0% and 45.5%, respectively, in terms of AUC. Overall, Algorithm 4.1 and LDSAM [134], provide the performance superior to that of the other methods and Algorithm 3.1 in terms of both the *precision score* and AUC.

4.5.2 Quantitative Evaluation on VOT2016

The average accuracy and robustness, and their rank are used to evaluate the performance of Algorithm 4.1 using VOT2016 benchmark dataset [94]. Table 4.3 shows the accuracy rank and overlap comparison of Algorithm 4.1 with that of the other tracking algorithms averaging over the VOT2016 sequences. Similarly, Table 4.4 shows the performance comparison of Algorithm 4.1 using robustness rank and failures averaging over the same challenging sequences. The last six columns of these two tables show the respective measures using different averaging methodologies, mean, weighted mean and pooled. The best three results are shown in **(red, bold)**, (violet, underline) and *(blue, italic)* fonts for better comparison of the proposed tracker with the other state-of-the-art trackers. Note that as the trackers with statistically equivalent results are merged while ranking, the different trackers may have same accuracy rank and robustness rank [94]. From Table 4.3, it is observed that Algorithm 4.1, in terms of overlap, stands first for the attribute motion change, and stood second for the remaining attributes except illumination change and occlusion, where it stood third. Also, Algorithm 3.1_HOG ranks first in all the attributes except illumination

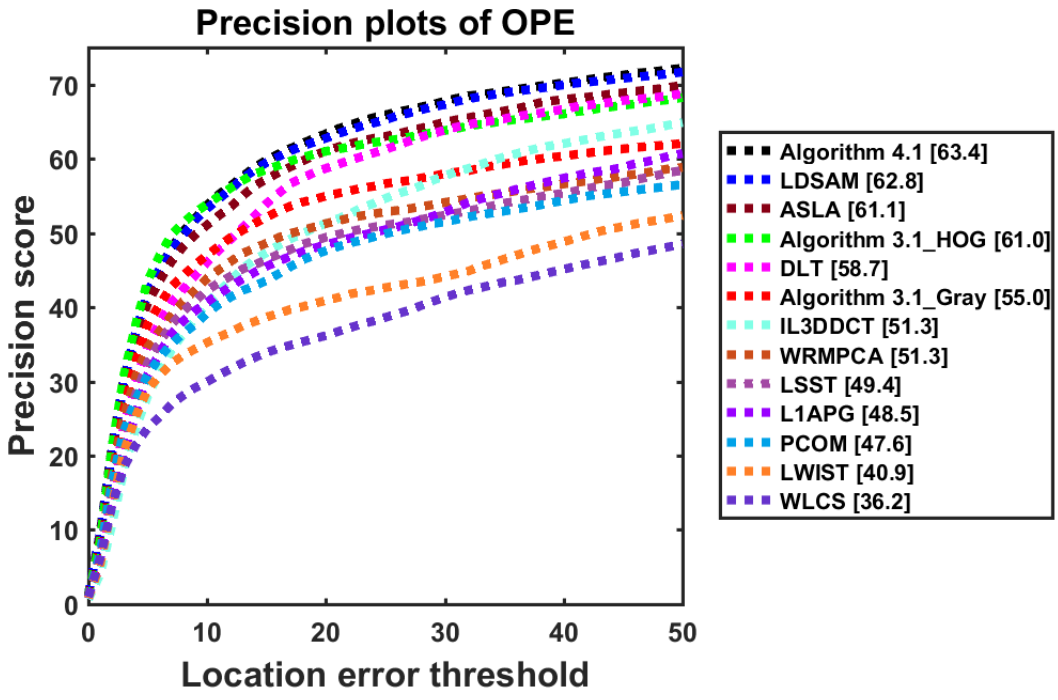


(a)

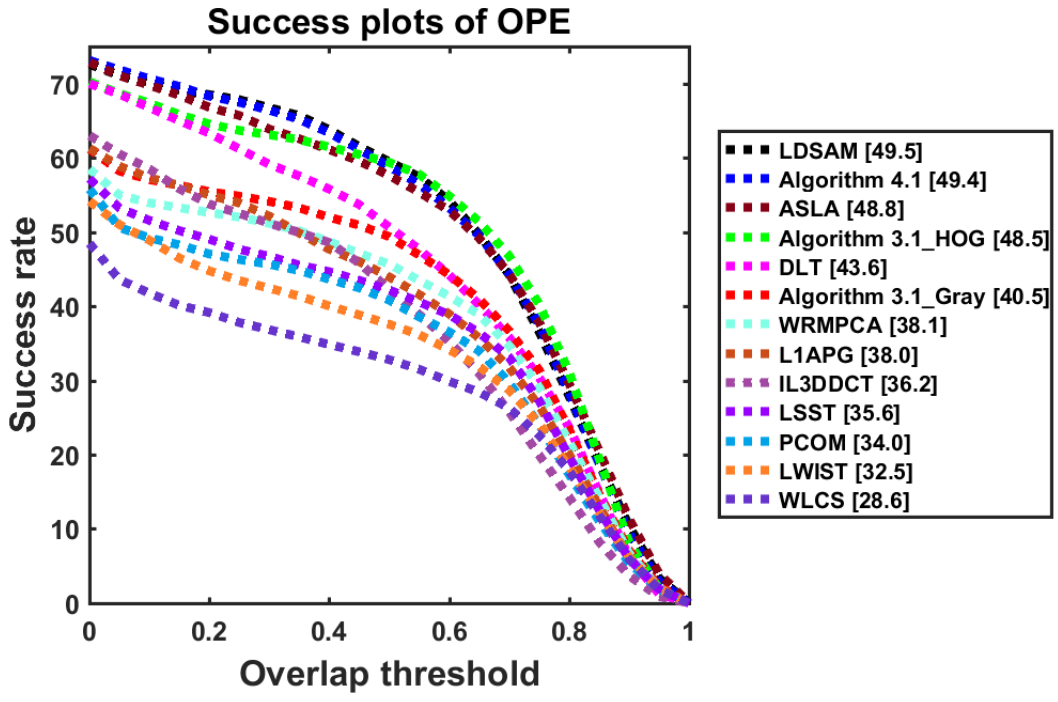


(b)

Figure 4.5: Performance evaluation of Algorithm 4.1 on OTB-50 sequences having occlusion using (a) the *precision plots* of OPE, where the *precision score* is shown along with the tracker name in the legend, and (b) the *success plots* of OPE, where AUC is shown along with the tracker name in the legend. The number 29 appearing in the title denotes the number of sequences associated with the occlusion attribute of OTB-50 dataset.



(a)



(b)

Figure 4.6: Overall performance evaluation of Algorithm 4.1 on OTB-50 using (a) the *precision plots* of OPE, where the *precision score* is shown along with the tracker name in the legend, and (b) the *success plots* of OPE, where AUC is shown along with the tracker name in the legend.

change and motion change. Further, ASLA stood third in camera motion and motion change attributes, whereas DLT stood second in motion change and occlusion, and third for the remaining attributes except camera motion and illumination change. For the sequences having illumination change, LWIST and WLCS stood first and second, respectively. Algorithm 4.1 has a performance better than that of ASLA in the face of all the challenging attributes. Also, the performance of Algorithm 4.1, in terms of overlap, has been improved to that of LDSAM [134] in all the challenging attributes due to the robust occlusion detection mechanism. It is also noticed that Algorithm 4.1 has a performance better or similar to that of Algorithm 3.1_HOG for all the challenging attributes except occlusion and size change. Further, Algorithm 4.1 outperforms all the methods for all the challenging attributes except illumination change, occlusion and size change.

Also, it is observed in Table 4.4 that Algorithm 4.1 ranks first in all the challenging attributes in terms of failures, except for motion change, where it stood second, and illumination change, where it stood fourth. It is also observed that Algorithm 3.1_HOG ranks third in size change. Further, ASLA stands first in illumination change and third in the remaining attributes, except camera motion, occlusion and size change, where it stood second, whereas DLT ranks first in motion change, third in camera motion and occlusion, and second in the remaining attributes, except illumination change and size change. For the sequences having illumination change, LWIST and PCOM stood second and third, respectively. Algorithm 4.1 has a performance superior to that of ASLA in face of all the challenging attributes, except illumination change. Also, the robust occlusion detection mechanism in Algorithm 4.1 has improved the performance in all the challenging attributes when compared to that of LDSAM [134]. In addition, Algorithm 4.1 has a performance superior to that of Algorithm 3.1 for all the challenging attributes. Also, Algorithm 4.1 exhibits a performance better than that of all the methods for all the challenging attributes of VOT2016, except illumination change and motion change.

The overall performance of Algorithm 4.1 is similar to that of Algorithm 3.1_HOG in terms of mean, weighted mean and pooled averaging of overlap. Also, the overall performance of Algorithm 4.1 is superior to that of the other methods in terms of mean, weighted mean and pooled averaging of overlap. On the other hand, the overall performance of Algorithm 4.1 is superior to that of the other methods and Algorithm 3.1_HOG in terms of mean, weighted mean and pooled averaging of failures.

4.5.3 Qualitative Evaluation

For qualitative evaluation of the trackers, some tracking results on a subset of the OTB-50 benchmark sequences are shown in Figure 4.7. In Figure 4.7, the tracking results of top twelve trackers, which are selected from Figure 4.6, on the six exemplar image frames are shown for each sequence and these six frames are selected at regular intervals without any bias. Algorithm 4.1 tracker successfully tracks the target in the all the frames of the *Doll*, *Faceocc2*, *Dudek*, *Fish*, *Girl*, *Freeman3*, *Jogging-2*, *Singer1*, *Walking2* and *Woman* sequences, which contain most of the real-time challenges such as pose change, partial occlusion, illumination change, scale change and out-of-plane rotation. This indicates the strong capabilities of Algorithm 4.1 in handling these challenges. Algorithm 4.1 tracker has slightly drifted away in middle few frames (between #250 to #415) of the *Girl* sequence and then starts tracking afterwards. It is also observed that LDSAM [134], performs better in all the sequences except *Walking2* and *Woman* sequences, where the sequence undergoes severe occlusion along with scale and appearance change, in spite of using local appearance model. Similarly, even with local appearance model, ASLA fails to track the object in *Faceocc2*, *Jogging-2*, *Walking2* and *Woman*, where the sequence undergoes partial or severe occlusion. These failures in both ASLA and LDSAM is because of the appearance model update with the imprecise tracked samples without removing the occlusion. It is also observed that Algorithm 3.1_HOG tracks the object completely in all the sequences except *Woman*, where it fails to track the object towards the end of the sequence, and *Girl*

Table 4.3: Accuracy rank (A-Rank) and average overlap comparison of Algorithm 4.1 with that of the compared trackers for different attributes of VOT2016. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Accuracy	camera motion		empty		illumination change		motion change		occlusion		size change		Mean		Weighted mean		Pooled	
Trackers	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap
Algorithm 4.1	**1.00**	0.45	**1.00**	0.52	**1.00**	*0.61*	**1.00**	**0.39**	**1.00**	*0.36*	**1.00**	0.42	**1.00**	0.46	**1.00**	0.45	**1.00**	0.46
LDSAM	7.00	0.33	12.00	0.36	12.00	0.31	7.00	0.28	**1.00**	0.31	*11.00*	0.25	8.33	0.31	8.33	0.31	11.00	0.33
Algorithm 3.1_HOG	**1.00**	**0.45**	**1.00**	**0.52**	**1.00**	0.61	**1.00**	0.38	**1.00**	**0.38**	**1.00**	**0.43**	**1.00**	**0.46**	**1.00**	**0.45**	**1.00**	**0.46**
Algorithm 3.1_Gray	**1.00**	0.43	**1.00**	0.48	*3.00*	0.54	**1.00**	0.38	**1.00**	0.34	**1.00**	0.41	*1.33*	0.43	*1.33*	0.43	**1.00**	0.43
ASLA	**1.00**	*0.44*	**1.00**	0.49	**1.00**	0.61	**1.00**	*0.38*	**1.00**	0.36	**1.00**	0.41	**1.00**	0.45	**1.00**	0.44	**1.00**	0.44
L1APG	**1.00**	0.43	**1.00**	0.50	11.00	0.45	**1.00**	0.37	**1.00**	0.35	4.00	0.33	3.17	0.40	3.17	0.41	**1.00**	0.42
WRMPCA	**1.00**	0.40	**1.00**	0.49	6.00	0.52	**1.00**	0.37	**1.00**	0.31	**1.00**	0.37	1.83	0.41	1.83	0.41	**1.00**	0.42
LWIST	**1.00**	0.42	**1.00**	0.49	**1.00**	**0.65**	**1.00**	0.34	**1.00**	0.34	**1.00**	0.41	**1.00**	0.44	**1.00**	0.42	**1.00**	0.43
WLCS	**1.00**	0.41	**1.00**	0.50	**1.00**	0.63	**1.00**	0.34	**1.00**	0.33	**1.00**	0.40	**1.00**	0.43	**1.00**	0.42	**1.00**	0.43
LSST	**1.00**	0.40	**1.00**	0.44	2.00	0.55	**1.00**	0.32	**1.00**	0.34	**1.00**	0.37	1.17	0.40	1.17	0.39	**1.00**	0.40
DLT	**1.00**	0.44	**1.00**	*0.50*	**1.00**	0.58	**1.00**	0.39	**1.00**	0.38	**1.00**	*0.42*	**1.00**	*0.45*	**1.00**	*0.44*	**1.00**	*0.45*
PCOM	**1.00**	0.41	**1.00**	0.49	*3.00*	0.57	**1.00**	0.36	**1.00**	0.36	**1.00**	0.41	*1.33*	0.43	*1.33*	0.42	**1.00**	0.43

Table 4.4: Robustness rank (R-Rank) and average failures comparison of Algorithm 4.1 with that of the compared trackers for different attributes of VOT2016. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Robustness	camera motion		empty		illumination change		motion change		occlusion		size change		Mean		Weighted mean		Pooled	
Trackers	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures
Algorithm 4.1	**1.00**	**90.87**	**1.00**	**40.53**	**1.00**	9.13	**1.00**	74.40	**1.00**	**28.27**	**1.00**	**33.73**	**1.00**	**46.16**	**1.00**	**57.59**	**1.00**	**192.80**
LDSAM	11.00	167.13	11.00	114.20	**1.00**	10.00	11.00	131.40	*4.00*	51.67	12.00	76.60	8.33	91.83	8.33	117.10	11.00	396.53
Algorithm 3.1_HOG	2.00	109.27	*5.00*	62.53	**1.00**	10.00	3.00	91.40	**1.00**	36.20	2.00	*42.73*	2.33	58.69	2.33	73.54	*4.00*	248.00
Algorithm 3.1_Gray	2.00	107.47	3.00	50.67	**1.00**	9.47	3.00	88.73	2.00	36.13	2.00	48.60	2.17	56.84	2.17	70.60	*4.00*	238.93
ASLA	**1.00**	92.80	3.00	*45.80*	**1.00**	**7.73**	3.00	*79.27*	2.00	33.93	2.00	40.20	*2.00*	*49.96*	*2.00*	*61.84*	2.00	*208.27*
L1APG	*4.00*	140.80	**1.00**	49.13	*4.00*	11.60	7.00	99.13	2.00	38.60	*5.00*	54.27	3.83	65.59	3.83	82.89	*4.00*	267.27
WRMPCA	*4.00*	106.60	3.00	52.47	2.00	10.47	3.00	91.93	*4.00*	38.20	2.00	48.33	3.00	58.00	3.00	71.51	*4.00*	242.60
LWIST	7.00	131.27	9.00	81.13	**1.00**	8.53	7.00	115.13	*4.00*	43.07	7.00	55.27	5.83	72.40	5.83	91.28	9.00	307.53
WLCS	9.00	164.73	11.00	89.53	**1.00**	9.13	9.00	124.13	6.00	46.67	9.00	60.87	7.50	82.51	7.50	105.86	11.00	355.27
LSST	8.00	135.60	9.00	77.87	*4.00*	11.40	7.00	115.20	*4.00*	45.60	10.00	61.47	7.00	74.52	7.00	93.11	9.00	315.20
DLT	**1.00**	*94.73*	**1.00**	42.20	**1.00**	10.07	**1.00**	**66.53**	2.00	*35.40*	2.00	43.33	1.33	48.71	1.33	60.06	**1.00**	201.47
PCOM	7.00	126.80	*5.00*	57.67	**1.00**	*8.87*	*6.00*	102.27	6.00	41.20	*5.00*	53.60	5.00	65.07	5.00	81.51	7.00	273.67

and *Jogging-2*, where it fails within initial few frames, whereas Algorithm 3.1_Gray tracks the object completely in all the sequences except *Girl* and *Woman*, where it fails within initial few frames, *Doll*, where it fails to track the object towards the end of the sequence. L1APG tracks the object completely in *Girl* and *Walking2* sequences and drifts away in *Faceocc2*, *Dudek* and *Fish* sequences in the last few frames. Further, WRMPCA fails to track the object completely in *Doll*, *Girl*, *Freeman3*, *Jogging-2* and *Woman* sequences. Further, LSST fails to estimate the scale and location of the object in most of the sequences except *Singer1* and *Freeman3* sequences. DLT tracks the object successfully in most of the sequences except *Doll*, *Girl* and *Jogging-2*. Finally, the overall qualitative performance of Algorithm 4.1 is better than that of the other methods and Algorithm 3.1.

4.6 Summary

In this chapter, a new tracking algorithm, Algorithm 4.1, which is based on a structural local 2DDCT sparse appearance model and an occlusion handling mechanism has been proposed. A procedure to reconstruct the holistic image from the overlapped local patches and then, a method to generate a robust occlusion map from the reconstructed holistic image have been given. The patch occlusion ratio has been defined and has been used in the confidence score computation by weighting the pooled features. Extensive experiments have been conducted on the two popular tracking benchmark datasets, OTB-50 and VOT2016, to analyze the performance of Algorithm 4.1. The quantitative and qualitative performance of Algorithm 4.1 has been compared with that of several state-of-the-art algorithms and Algorithm 3.1 using these benchmark datasets. Algorithm 4.1 outperforms Algorithm 3.1 and other methods in terms of the *precision* and *success plots* averaged over the OTB-50 benchmark sequences. Also, the performance of Algorithm 4.1 is similar to that of Algorithm 3.1_HOG and generally superior to that of the other considered methods in

terms of overlap for VOT2016 dataset. Further, Algorithm 4.1 generally outperforms Algorithm 3.1 and other methods in terms of failures for VOT2016 dataset. Even though Algorithm 4.1 generally performs better than that of Algorithm 3.1 and other methods, it still needs an improvement in motion blur, fast motion, in-plane rotation, out-of-view and background clutter challenging attributes of OTB-50, and illumination change, motion change and occlusion attributes of VOT2016. To improve the tracking performance on some of these challenges, a third tracking algorithm based on a collaboration of the discriminative and generative appearance models is presented in the next chapter.

Figure 4.7: Examples of tracking results of the compared methods on the ten OTB-50 benchmark sequences.

Chapter 5

Collaboration of Kernelized Correlation Filters and the Bilateral 2DPCA Subspace for Visual Tracking

5.1 Introduction

Algorithms 3.1 and 4.1 proposed in the preceding chapters do not perform well in the challenging attributes of motion blur (MB) and fast motion (FM) of the object tracking benchmark-50 (OTB-50) dataset, and illumination change, motion change and occlusion of the visual object tracking 2016 (VOT2016) dataset. These two proposed algorithms are based on the particle filter framework. The main drawback of the particle filter-based tracking algorithms is that they cannot effectively track a fast-moving object with random velocity and acceleration [151]. Also, in order to cover the search range and object states well, the particle filter has to densely generate probable target locations in a wide range of area resulting in a large number of candidates, and hence, increasing the complexity of the tracking algorithm. As can be seen from Chapter 2, the target state $\mathbf{s}_t$ is modeled by an affine transformation with six parameters x_t, y_t, θ_t, s_t, α_t and ϕ_t. Among the variances of these six parameters, the variances of translations, x_t and y_t, are responsible mainly for the search range of

the candidates. Assuming a larger or a smaller variance of translations results in a loss of the target during slow and fast motion of the target, respectively.

In order to address this issue and to improve the performance of the tracker in view of some of the challenges discussed above, in this chapter, a new tracking algorithm based on a collaboration of the discriminative and generative appearance models is proposed [152]. In the discriminative model, two kernelized correlation filters (KCFs) are used to estimate the location (x_t, y_t) of the target, and then, a new generative model is used to find the other affine motion parameters (θ_t, s_t, α_t, ϕ_t) of the target. The above method of finding the location and other affine motion parameters of the target is based on the ideas that (1) the discriminative capability of a tracker plays an important role in estimating the location of the target rather than in finding the other affine motion parameters of the target and (2) the generative capability of a tracker plays an important role in estimating the other affine motion parameters of the target. In the generative model, the robust coding technique used in the first tracking algorithm is extended to two dimensions, and then used in the bilateral two dimensional principal component analysis (B2DPCA) reconstruction procedure to develop an iteratively reweighted robust coding (IRRC) technique [118]. To find the similarity between the candidate and its reconstructed sample from the B2DPCA subspace, a two dimensional robust coding (2DRC) distance measure is defined, which in turn is used to calculate the observation likelihood. The occlusion information captured by the weights, which are obtained from IRRC, are used to generate an occlusion map. The occlusion map thus generated is used to develop a mechanism for the observation model update of both the B2DPCA subspace and the two KCFs. Experiments conducted on the two popular benchmark datasets and comparison with the state-of-the-art methods bear out the competency and effectiveness of the proposed algorithm for visual tracking.

This chapter is organized as follows. KCFs are discussed in Section 5.2. Object representation based on the B2DPCA projection matrices and 2D robust coding is

explained in Section 5.3. The proposed tracking algorithm is developed in Section 5.4. Experimental results obtained using the two popular benchmark datasets are discussed in Section 5.5 followed by a summary of this chapter in Section 5.6.

5.2 Kernelized Correlation Filters

The correlation filter-based visual tracking has achieved a great success due to its high efficiency and performance, and hence, has attracted much attention among research community [67, 68, 71, 80]. Notable among them is, the kernelized correlation filters (KCF) [71] that employ numerous negative samples to enhance the discriminative capability of the tracking-by-detection scheme by exploiting the structure of the circulant matrix for computational efficiency. In KCF, the object appearance is modeled using a correlation filter $\mathbf{H}$ trained on an image patch $\mathbf{X}$ of $d_h \times d_w$ pixels, where all the circular shifts of $\mathbf{X}_{i,j}$ with $(i, j) \in \{0, 1, ..., d_h - 1\} \times \{0, 1, ..., d_w - 1\}$ are generated as training samples with Gaussian function label $\mathbf{g}_{i,j}$. The optimal filter weights $\mathbf{H}$ are obtained by minimizing the following objective function

$$\mathbf{H} = \arg\min_{\mathbf{H}} \sum_{i,j} |\langle \Phi(\mathbf{X}_{i,j}), \mathbf{H} \rangle - \mathbf{g}_{i,j}|^2 + \xi \|\mathbf{H}\|^2 \tag{5.1}$$

where Φ denotes the mapping to a kernel space and ξ is a regularization parameter. By using the Fourier transform, the objective function in (5.1) is minimized as $\mathbf{H} = \sum_{i,j} \boldsymbol{\alpha}_{i,j} \Phi(\mathbf{X}_{i,j})$, and the coefficient $\boldsymbol{\alpha}$ is given by

$$\boldsymbol{\alpha} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathbf{g})}{\mathcal{F}\left(\langle \Phi(\mathbf{X}), \Phi(\mathbf{X}) \rangle\right) + \xi} \right) \tag{5.2}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote, respectively, the Fourier transform and its inverse. The learned coefficients model $\hat{\boldsymbol{\alpha}}$ and the target appearance model $\hat{\mathbf{X}}$ along with an image

patch $\mathbf{X}^D$ cropped out in the new frame are used to find the response map as

$$\overline{\mathbf{g}} = \mathcal{F}^{-1}\left(\mathcal{F}(\hat{\boldsymbol{\alpha}}) \odot \mathcal{F}\left(\langle \Phi(\mathbf{X}^D), \Phi(\hat{\mathbf{X}}) \rangle\right)\right) \tag{5.3}$$

where $\odot$ is the Hadamard product. The maximal value of the response map $\overline{\mathbf{g}}$ gives the position of the target. Despite its good performance, there are still some issues, such as scale variations and out-of-view problems, that need to be addressed.

5.3 Object Representation based on Bilateral 2DPCA Projection Matrices and 2D Robust Coding

In incremental visual tracking (IVT) [1], a low dimensional principal component analysis (PCA) subspace is used to represent the object, where PCA subspace is learned and updated efficiently to adapt the appearance variations of the object. Further, incremental B2DPCA has been used to model the object appearance in visual tracking based on the maximum likelihood estimation (MLE) [63]. Wang *et al.* have exploited the strength of both the subspace and sparse representations in [3] and [19] by introducing l_1-regularization into the PCA and B2DPCA reconstruction, respectively. Even though the subspace-based trackers [1, 63] are robust to in-plane rotation, illumination variation, scale change and pose change, they are sensitive to partial occlusion due to their underlying assumption that the residual is Gaussian distributed with small variances. This is not valid as the residual cannot be modeled with small variances during partial occlusion. On the other hand, if the residual is assumed to be Laplacian distributed, then it aims to handle outliers. As the residual cannot be modeled with neither Gaussian nor Laplacian distribution during occlusion in visual tracking, the strengths of both the B2DPCA subspace representation and RSC were exploited in [64] by introducing l_1-regularization into the B2DPCA reconstruction. In [64], the object appearance is represented by the B2DPCA projection

matrices and its observation model update mechanism is similar to that in [54]. The drawback of the appearance model in [64] is that the unnecessary imposition of l_1-norm constraint on the projection coefficients even though they are not sparse due to the orthogonality of the B2DPCA projection matrices.

In this section, to account for non-Gaussian or non-Laplacian residual/noise, robust coding, discussed in Section 3.2.1, is extended to two dimensions, and then used in the B2DPCA reconstruction procedure to model the object appearance. The object appearance is modeled by two separate B2DPCA projection matrices as in (2.2) and the projection coefficient $\mathbf{Z}$ is computed using (2.1), where $\overline{\mathbf{Y}} \in \mathbb{R}^{d_l \times d_r} = \mathbf{Y} - \boldsymbol{\mu}_{BP}$ represents the centered image observation matrix, $\boldsymbol{\mu}_{BP} \in \mathbb{R}^{d_l \times d_r}$ represents mean matrix, $\mathbf{U}_{BP} \in \mathbb{R}^{d_l \times k_l}$ and $\mathbf{V}_{BP} \in \mathbb{R}^{d_r \times k_r}$ represent orthogonal left- and right-projection matrices, respectively, $\mathbf{Z} \in \mathbb{R}^{k_l \times k_r}$ denotes the projection coefficients, $d_l \times d_r$ the size of the observation matrix, and k_l and k_r are the number of B2DPCA left- and right-projection basis vectors, respectively. As the target templates are coherent in [2, 13], the coding coefficients are sparse and hence, there is a requirement of l_1-norm constraint on the coding coefficients. But in the proposed appearance model, the projection coefficients are not sparse due to the orthogonality of the B2DPCA projection matrices and hence, it is not required to impose complex l_1-norm constraint on the projection coefficients. This is in contrast to [64], where the unnecessary complex l_1-norm constraint is imposed on the projection coefficients $\mathbf{Z}$ in spite of using the B2DPCA projection matrices $(\mathbf{U}_{BP}, \mathbf{V}_{BP})$.

Expressing the left- and right-projection matrices as $\mathbf{U}_{BP} = [\mathbf{u}_1; \mathbf{u}_2; ...; \mathbf{u}_{d_l}]$ and $\mathbf{V}_{BP} = [\mathbf{v}_1; \mathbf{v}_2; ...; \mathbf{v}_{d_r}]$ respectively, where the vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ are the i-th and j-th rows of $\mathbf{U}_{BP}$ and $\mathbf{V}_{BP}$, respectively, and denoting the coding residual error matrix as $\mathbf{E} = \overline{\mathbf{Y}} - \mathbf{U}_{BP}\,\mathbf{Z}\,\mathbf{V}_{BP}^{\mathsf{T}}$, each element of the residual error matrix $\mathbf{E}$ is written as $e_{ij} = \overline{y}_{ij} - \mathbf{u}_i\,\mathbf{Z}\,\mathbf{v}_j^{\mathsf{T}}$. Assume that the residuals $e_{11}, ..., e_{d_l d_r}$ are independently and identically distributed (i.i.d) according to some probability density function $f_\Theta(e_{ij})$, where Θ denotes the parameter set that characterizes the probability distribution. Then,

maximizing the likelihood function $L_\Theta(e_{11}, ..., e_{d_l d_r}) = \prod_{j=1}^{d_r} \prod_{i=1}^{d_l} f_\Theta(e_{ij})$ is equivalent to minimizing the objective function: $-\ln L_\Theta(e_{11}, ..., e_{d_l d_r}) = \sum_{j=1}^{d_r} \sum_{i=1}^{d_l} \rho_\Theta(e_{ij})$, where $\rho_\Theta(e_{ij}) = -\ln f_\Theta(e_{ij})$. From this discussion, MLE of $\mathbf{Z}$, referred to as 2D robust coding (2DRC), can be formulated as the following minimization problem

$$\min_{\mathbf{Z}} \sum_{j=1}^{d_r} \sum_{i=1}^{d_l} \rho_\Theta(e_{ij}) = \min_{\mathbf{Z}} \sum_{j=1}^{d_r} \sum_{i=1}^{d_l} \rho_\Theta(\overline{y}_{ij} - \mathbf{u}_i \, \mathbf{Z} \, \mathbf{v}_j^\mathsf{T}) \tag{5.4}$$

The introduction of 2DRC takes care of non-Gaussian or non-Laplacian noise and avoids the effect of outliers (e.g., occluded or corrupted pixels) while computing the projection coefficients of the B2DPCA projection matrices. Now, MLE of $\mathbf{Z}$ can be obtained by solving (5.4), but the problem is as to how to find the distribution ρ_Θ (or f_Θ). Explicitly taking f_Θ as Gaussian or Laplacian distribution is simple, but not effective during occlusion. Based on the assumptions on ρ_Θ specified in [115], the above minimization problem is transformed into a weighted least squares (WLS) problem, given by

$$\min_{\mathbf{Z}} \|\mathbf{A}^{\frac{1}{2}} \odot (\overline{\mathbf{Y}} - \mathbf{U}_{BP} \, \mathbf{Z} \, \mathbf{V}_{BP}^\mathsf{T})\|_F^2 \tag{5.5}$$

where $\mathbf{A} \in \mathbb{R}^{d_l \times d_r}$ is a weight matrix to model different types of noise, and its element a_{ij} is the weight assigned to each pixel of the observed image sample $\mathbf{Y}$ depending on the value of the residual e_{ij}, and $\odot$ is the Hadamard product. Since the weight matrix $\mathbf{A}$ is unknown and needs to be estimated, WLS in (5.5) is a local approximation of 2DRC in (5.4). Therefore, the 2DRC minimization procedure can be converted into an IRRC problem with $\mathbf{A}$ being updated using the residuals in the previous iteration. Since the distribution ρ_Θ is unknown, it is difficult to find the weight matrix $\mathbf{A}$. Thus, a logistic function given by

$$a_{ij} = \frac{\exp\left(\delta_{BP}\left[\beta_{BP} - e_{ij}^2\right]\right)}{1 + \exp\left(\delta_{BP}\left[\beta_{BP} - e_{ij}^2\right]\right)}, \tag{5.6}$$

where δ_{BP} controls the decreasing rate from 1 to 0, and β_{BP} controls the location of the demarcation point, is chosen as the weight function, as it satisfies the following properties: (1) weight assigned to each pixel of the observed image $\mathbf{Y}$ depends on the corresponding value of the residual $\mathbf{E}$ and (2) the weight function has higher capability to classify inliers and outliers [115]. This weight function is bounded in [0,1] and adaptively assigns low weights to the outliers (usually the pixels with large residuals) to reduce their effect on the estimation of the projection coefficients $\mathbf{Z}$ so that the sensitivity to outliers can be greatly reduced. Even though the tracking methods in [53], [54], and [64] use RSC, they differ from the proposed algorithm in a number of ways. The tracking methods in [53] and [54] use a target template-based appearance model, and hence, use l_1-norm constraint on the coding coefficients, and are computationally complex. Also, they differ in the way the observation model is updated. Even though the appearance model of [64] is based on B2DPCA, it imposes l_1-norm constraint on the projection coefficients thereby increasing the computational complexity. Further, its weight function and the observation model update mechanism are different from that of the proposed algorithm.

The minimization problem in (5.5) can be solved by estimating iteratively the weight matrix $\mathbf{A}$ using (5.6) and the projection coefficients $\mathbf{Z}_{opt}$ using the following equation

$$\mathbf{Z}_{opt} = \mathbf{U}_{BP}^{\mathsf{T}} \left(\mathbf{A} \odot \overline{\mathbf{Y}}\right) \mathbf{V}_{BP} \tag{5.7}$$

The estimation of $\mathbf{A}$ and $\mathbf{Z}_{opt}$ recursively is referred to as iteratively reweighted robust coding (IRRC) technique and is given in Procedure 5.1. This is terminated when the following criterion is satisfied:

$$\frac{\|\mathbf{A}^q - \mathbf{A}^{q-1}\|_F}{\|\mathbf{A}^{q-1}\|_F} < \psi_{BP}\,, \tag{5.8}$$

where ψ_{BP} is a small positive scalar constant.

Procedure 5.1 Computation of $\mathbf{Z}_{opt}$ and $\mathbf{A}$ by the iteratively reweighted robust coding technique

Input: Centered image observation matrix $\overline{\mathbf{Y}}$, left- and right-projection matrices $\mathbf{U}_{BP}$ and $\mathbf{V}_{BP}$, previous weight matrix $\mathbf{A}_{t-1}$ corresponding to the tracking result at time $t-1$, constants δ_{BP} and β_{BP}

1: Initialize $q = 0$ and $\mathbf{A}^q = \mathbf{A}_{t-1}$

2: Compute basis coefficients $\mathbf{Z}^q = \mathbf{U}_{BP}^{\top}(\mathbf{A}^q \odot \overline{\mathbf{Y}})\,\mathbf{V}_{BP}$

3: Iterate

4: $\quad q \leftarrow q + 1$

5: $\quad$ Compute residual $\mathbf{E}^q = \overline{\mathbf{Y}} - \mathbf{U}_{BP}\,\mathbf{Z}^{q-1}\,\mathbf{V}_{BP}^{\top}$

6: $\quad$ Compute the weights using

$$a_{ij}^q = \frac{\exp\left(\delta_{BP}\left[\beta_{BP} - \left(e_{ij}^q\right)^2\right]\right)}{1 + \exp\left(\delta_{BP}\left[\beta_{BP} - \left(e_{ij}^q\right)^2\right]\right)}; \quad \begin{matrix} i = 1, 2, .., d_l \\ j = 1, 2, .., d_r \end{matrix}$$

7:

8: $\quad$ Recompute $\mathbf{Z}^q = \mathbf{U}_{BP}^{\top}(\mathbf{A}^q \odot \overline{\mathbf{Y}})\mathbf{V}_{BP}$

9: Until convergence or termination

Output: Basis coefficients $\mathbf{Z}_{opt}$, weight matrix $\mathbf{A}$

5.4 Proposed Tracking Algorithm

Most of the collaborative methods [43–46] find all the affine motion parameters of the target by combining the individual scores of both the generative and discriminative models, whereas the proposed algorithm explores the discriminative model to estimate the target location (x_t, y_t) and the particle filter-based generative model to find the remaining affine parameters of the target $(\theta_t, s_t, \alpha_t, \phi_t)$. This is based on the intuition that the discriminative capability of a tracker plays a prominent role in estimating the location of the target rather than in finding the other affine motion parameters of the target. On the other hand, the generative capability of a tracker plays a prominent role in finding the other affine motion parameters of the target. The block diagram of the proposed tracking algorithm is shown Figure 5.1.

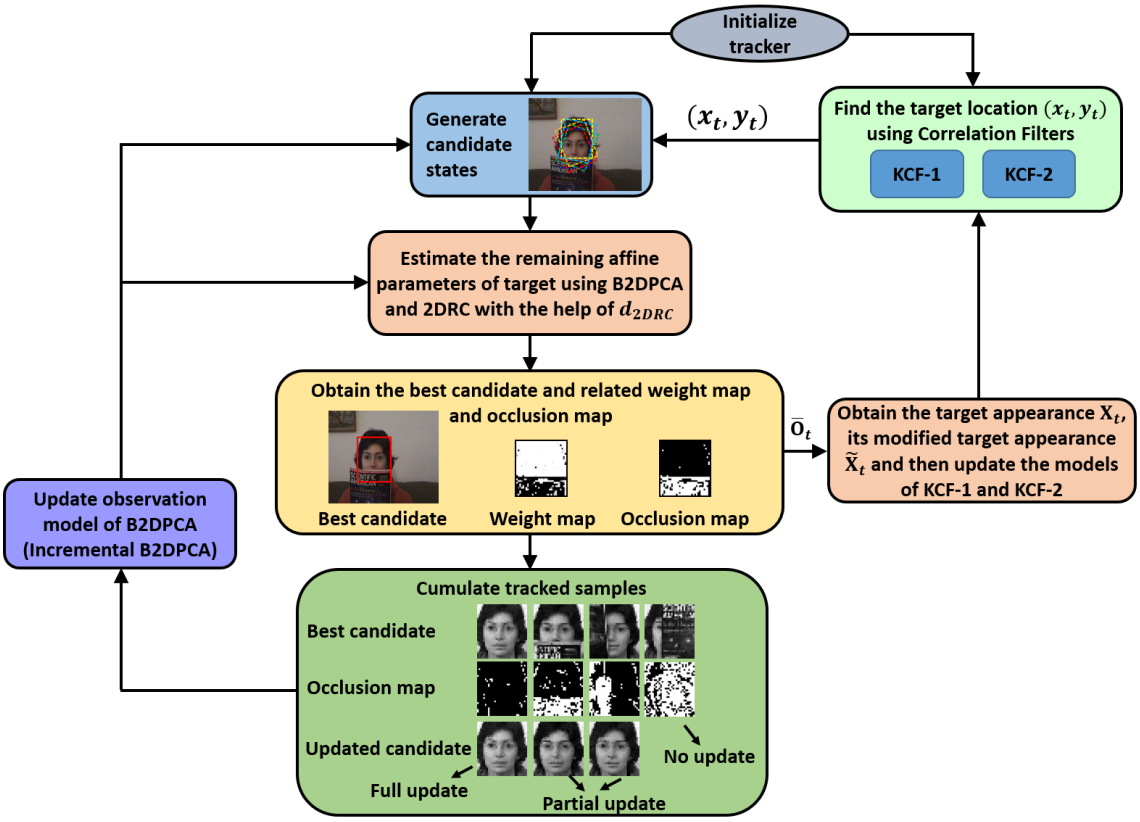


Figure 5.1: Block diagram of the proposed tracking algorithm.

5.4.1 Target Location Estimation using Kernelized Correlation Filters

For target location estimation, it is proposed to use two kernelized correlation filters, KCF-1 and KCF-2, each with its own target appearance model $\hat{\mathbf{X}}_t^k$ and learned coefficients model $\hat{\boldsymbol{\alpha}}_t^k$, where $k = 1, 2$ indicates which correlation filter the model belongs to. An image patch $\mathbf{X}_t^D$ of new window size, which is estimated in the previous frame $t-1$, is cropped out from the previous target position (x_{t-1}, y_{t-1}) in the current frame t, and then resized to the initial window size in order to preserve the consistency of the object representation in the scale space. This patch along with their respective models $\hat{\mathbf{X}}_{t-1}^k$ and $\hat{\boldsymbol{\alpha}}_{t-1}^k$ is used to find the response maps $\overline{\mathbf{g}}_t^k$ of the two correlation

filters using (5.3). The resulting response maps $\overline{\mathbf{g}}_t^k$ are energy normalized to scale the peak value according to the total energy in the respective response map. This helps to normalize low/high peak values when the entire response map is low/high due to the image characteristics such as illumination [153]. Finally, the location of the maximum value of the response map $\widetilde{\mathbf{g}}_t$ that is computed employing (5.9) is used to find the position of the target (x_t, y_t).

$$\widetilde{\mathbf{g}}_t = \begin{cases} \overline{\mathbf{g}}_t^1, & \text{if } \max(\overline{\mathbf{g}}_t^1) > \max(\overline{\mathbf{g}}_t^2) \\ \overline{\mathbf{g}}_t^2, & \text{otherwise} \end{cases} \tag{5.9}$$

5.4.2 Target State Estimation using Bilateral 2DPCA and 2DRC

It is assumed that an image observation sample $\mathbf{Y}$ can be generated from a B2DPCA subspace of the target object spanned by the B2DPCA basis vectors $(\mathbf{U}_{BP}, \mathbf{V}_{BP})$ and centered at mean $\boldsymbol{\mu}_{BP}$. As opposed to the trivial templates used in appearance models of [3, 65, 154], the proposed appearance model uses the weight matrix $\mathbf{A}$ to model the occlusion/outliers. Hence, even during occlusion, the weighted image observation can be represented by a weighted sum of the mean $\boldsymbol{\mu}_{BP}$ and the linear combination of the B2DPCA basis vectors $(\mathbf{U}_{BP}, \mathbf{V}_{BP})$, as in (5.10), giving less importance to the outliers/occluded pixels in the representation as well as in the estimation of the projection coefficients $\mathbf{Z}$.

$$\mathbf{A}^{\frac{1}{2}} \odot \mathbf{Y} = \mathbf{A}^{\frac{1}{2}} \odot \left(\boldsymbol{\mu}_{BP} + \mathbf{U}_{BP}\, \mathbf{Z}\, \mathbf{V}_{BP}^{\mathsf{T}} \right) \tag{5.10}$$

In addition to an accurate estimation of the projection coefficients $\mathbf{Z}$, visual tracking also needs a distance metric to find the similarity between a noisy observation and its reconstructed sample from the dictionary or the subspace [1, 65, 116]. In general, the distance metric is inversely proportional to the maximum joint likelihood with

respect to the projection coefficients $\mathbf{Z}$ [1, 65],

$$d(\mathbf{Y}; \mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP}) \propto -\ln \max_{\mathbf{Z}} p(\mathbf{Y}, \mathbf{Z}) \propto -\ln \max_{\mathbf{Z}} p(\mathbf{Y}|\mathbf{Z})\, p(\mathbf{Z})$$

Assuming a uniform prior, the distance metric is written as

$$d(\mathbf{Y}; \mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP}) \propto -\ln \max_{\mathbf{Z}} \exp\left(-\frac{1}{2}\|\overline{\mathbf{Y}} - \mathbf{U}_{BP}\,\mathbf{Z}\,\mathbf{V}_{BP}^{\top}\|_F^2\right)$$

This distance metric is sensitive to occlusion/outliers as it considers the occluded/outlier pixels for the similarity measurement. In order to make the distance metric robust to occlusion/outliers, it should give less importance to the reconstruction error due to the occluded pixels or outliers, and hence, in this work, a new 2DRC distance, defined as

$$d_{2DRC}(\mathbf{Y}; \mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP}) = \|\mathbf{A}^{\frac{1}{2}}\,\mathbf{E}\|_F^2 + \lambda_{BP}\|1 - \mathbf{A}^{\frac{1}{2}}\|_F^2 \tag{5.11}$$

where $\mathbf{E} = \overline{\mathbf{Y}} - \mathbf{U}_{BP}\,\mathbf{Z}\,\mathbf{V}_{BP}^{\top}$ and λ_{BP} is a penalty constant, is proposed.

In visual tracking based on Bayesian inference framework, the confidence of each particle is given by its observation likelihood, and in the proposed algorithm, it is defined as

$$p(\mathbf{Y}_t^m|\mathbf{s}_t^m) = \exp\left(-\frac{1}{\gamma}\, d_{2DRC}(\mathbf{Y}_t^m; \mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP})\right) \tag{5.12}$$

where γ is a constant. As the observation likelihood in the proposed algorithm considers the distance metric $d_{2DRC}(\mathbf{Y}; \mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP})$, the effect of occlusion/outliers on the likelihood is reduced thereby making the likelihood robust to occlusion/outliers.

For each candidate image sample $\mathbf{Y}_t^m$ observed by a state $\mathbf{s}_t^m$, the minimization problem in (5.5) is solved efficiently by the IRRC technique as per Procedure 5.1 to obtain $\mathbf{Z}_{t,opt}^m$ and $\mathbf{A}_t^m$. Now among the candidate samples, the optimal state of the tracked target $\hat{\mathbf{s}}_t$ is found using (5.11), (5.12) and (2.4). Finally, the observation models are adapted to handle the appearance change of the target by incrementally updating both the KCF models ($\hat{\mathbf{X}}_t^1$, $\hat{\boldsymbol{\alpha}}_t^1$, $\hat{\mathbf{X}}_t^2$, $\hat{\boldsymbol{\alpha}}_t^2$), and the B2DPCA subspace model

$(\mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP})$, as discussed in the next subsection.

5.4.3 Observation Model Update

The update of the observation model is very much essential to handle the appearance variations of the object, but the update with imprecise samples will cause tracking drift due to the model degradation. Therefore, the imprecise samples should be avoided during the model update.

a) Bilateral 2DPCA

The appearance variations of the object are handled by incrementally updating the B2DPCA projection matrices $(\mathbf{U}_{BP}, \mathbf{V}_{BP})$, and the mean $\boldsymbol{\mu}_{BP}$. In order to avoid update with imprecise samples having occlusion/outliers, it is very important to extract the occlusion/outliers information from the tracking results. As the occluded/outlier pixels have low weights in the proposed algorithm, the occlusion/outliers information is extracted from the weight matrix $\mathbf{A}$ of the tracked target candidate. But in [3, 19], the occlusion/outliers information is extracted from the coefficients of the trivial templates. In the proposed observation model update, a pixel is considered either noisy or occluded, if the corresponding weight $a_{ij} < 0.5$ while generating a binary occlusion map $\mathbf{O}_t$. Here, the weight $a_{ij} < 0.5$ is considered without any bias for the detection of outliers/occlusion as the weight values are bounded in [0,1] and 0.5 is midpoint of this range. Note that no arbitrary threshold is used to detect the occlusion/outlier as in other methods. Using the weight matrix $\mathbf{A}$, the occlusion map $\mathbf{O}_t$, with an entry of unity indicating outlier and an entry of zero indicating inlier pixel, is generated according to the following rule:

$$\mathbf{O}_t(i,j) = \begin{cases} 1, & \text{if } a_{ij} < 0.5 \qquad i = 1, 2, .., d_l \\ 0, & \text{otherwise} \qquad j = 1, 2, .., d_r \end{cases} \tag{5.13}$$

Usually the occluded region is a large connected area compared to that of random noises or object appearance variations, which are comparatively very small. Hence, to retain the large connected area, and to fill the small hole between the regions and to remove the small regions, morphological operations and connected component analysis are performed on the occlusion map. This updated occlusion map $\hat{\mathbf{O}}_t$ is used to find the occlusion ratio τ, which is the ratio of the number of ones in $\hat{\mathbf{O}}_t$ to the total number of elements in $\hat{\mathbf{O}}_t$. Now, with the help of two thresholds, τ_1 and τ_2, the occlusion ratio τ is used to decide whether the tracked sample is utilized fully, or partially, or not utilized at all, in the observation model update. In the absence of occlusion (when $\tau < \tau_1$), the tracked sample is used directly for the model update (full update). During a partial occlusion (when $\tau_1 < \tau < \tau_2$), the occluded pixels in the tracked sample are replaced with the corresponding pixels from the previously updated mean $\boldsymbol{\mu}_{BP}$ to get an updated sample, which is free from occlusion, and is used in a model update. Otherwise, the tracked sample is not used for the model update due to severe occlusion (when $\tau > \tau_2$). These updated new observations are accumulated and used to update the observation model ($\mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP}$) by incremental subspace learning [63].

b) Kernelized Correlation filters

In the proposed algorithm, the model of each kernelized correlation filter consists of its own target appearance $\hat{\mathbf{X}}_t^k$ and the learned coefficients model $\hat{\boldsymbol{\alpha}}_t^k$, where $k = 1, 2$ indicates which correlation filter the model belongs to. In order to preserve the consistency of the object representation in the scale space, the optimal state $\hat{\mathbf{s}}_t$ of the tracked target, obtained from (2.4), is used to find the new target and window sizes, and then an image is cropped out from the current frame corresponding to the new window size and position $(\hat{x}_t, \hat{y}_t)$, and it is resized to the initial window size to obtain the target appearance $\mathbf{X}_t$. The model of the first correlation filter KCF-1 is updated

by the linear interpolation, given by

$$\mathcal{F}(\hat{\boldsymbol{\alpha}}_t^1) = \begin{cases} (1-\eta)\mathcal{F}(\hat{\boldsymbol{\alpha}}_{t-1}^1) + \eta\mathcal{F}(\boldsymbol{\alpha}_t^1), & \text{if } \tau \leq \tau_{KCF} \\ \mathcal{F}(\hat{\boldsymbol{\alpha}}_{t-1}^1), & \text{otherwise} \end{cases}$$
$$\mathcal{F}(\hat{\mathbf{X}}_t^1) = \begin{cases} (1-\eta)\mathcal{F}(\hat{\mathbf{X}}_{t-1}^1) + \eta\mathcal{F}(\mathbf{X}_t), & \text{if } \tau \leq \tau_{KCF} \\ \mathcal{F}(\hat{\mathbf{X}}_{t-1}^1), & \text{otherwise} \end{cases} \tag{5.14}$$

where $\boldsymbol{\alpha}_t^1$ is the learned coefficient obtained from (5.2) using the target appearance $\mathbf{X}_t$ at time t, η is the learning rate parameter and τ_{KCF} is a threshold. Note that both the target appearance model $\hat{\mathbf{X}}_t^1$ and the learned coefficients model $\hat{\boldsymbol{\alpha}}_t^1$ of KCF-1 are not updated when the occlusion ratio $\tau > \tau_{KCF}$. This prevents the model from getting degraded during severe occlusion, and hence, the tracking drift. Further, with the help of occlusion map $\hat{\mathbf{O}}_t$ and the previous target appearance model $\hat{\mathbf{X}}_{t-1}^2$ of KCF-2, the modified target appearance $\tilde{\mathbf{X}}_t$ is obtained from the target appearance $\mathbf{X}_t$ as

$$\tilde{\mathbf{X}}_t = \overline{\mathbf{O}}_t \odot \hat{\mathbf{X}}_{t-1}^2 + (1 - \overline{\mathbf{O}}_t) \odot \mathbf{X}_t \tag{5.15}$$

where $\overline{\mathbf{O}}_t$ is the resized occlusion map obtained from $\hat{\mathbf{O}}_t$ to match the matrix dimensions of $\hat{\mathbf{X}}_{t-1}^2$. Note that from Figure 5.2 the modified target appearance $\tilde{\mathbf{X}}_t$ is free from the occlusion only inside the target region but not outside, since the occlusion maps $\mathbf{O}_t$ and $\hat{\mathbf{O}}_t$ are obtained only for the target region in a generative appearance model of B2DPCA. Now, the models of KCF-2 are updated as

$$\begin{aligned} \mathcal{F}(\hat{\boldsymbol{\alpha}}_t^2) &= (1-\eta)\mathcal{F}(\hat{\boldsymbol{\alpha}}_{t-1}^2) + \eta\mathcal{F}(\boldsymbol{\alpha}_t^2) \\ \mathcal{F}(\hat{\mathbf{X}}_t^2) &= (1-\eta)\mathcal{F}(\hat{\mathbf{X}}_{t-1}^2) + \eta\mathcal{F}(\tilde{\mathbf{X}}_t) \end{aligned} \tag{5.16}$$

where $\boldsymbol{\alpha}_t^2$ is the learned coefficients obtained from the modified target appearance $\tilde{\mathbf{X}}_t$ using (5.2). By employing the modified target appearance $\tilde{\mathbf{X}}_t$ instead of $\mathbf{X}_t$ in both the learned coefficients $\boldsymbol{\alpha}_t^2$ computation and the target appearance model $\hat{\mathbf{X}}_t^2$ update,

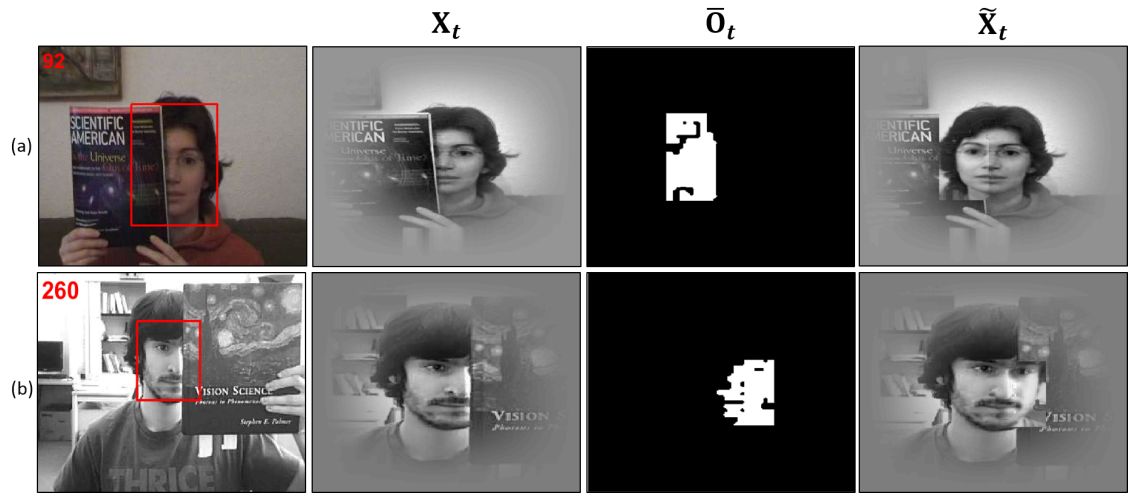


Figure 5.2: Some representative cases of (a) *Faceocc1* (#92) and (b) *Faceocc2* (#260) sequences showing the target appearance $\mathbf{X}_t$, the resized occlusion map $\overline{\mathbf{O}}_t$ and the modified target appearance $\tilde{\mathbf{X}}_t$.

the models of KCF-2 are prevented from degradation due to occlusion.

Algorithm 5.1 shows the steps of the proposed tracking method based on KCF and B2DPCA. To understand the advantages of using two KCFs on the tracking performance, two special cases of Algorithm 5.1 are considered each with only one KCF enabled and the other disabled. The special case of Algorithm 5.1 that uses only KCF-1, whose target appearance model $\hat{\mathbf{X}}_t^1$ and learned coefficients model $\hat{\boldsymbol{\alpha}}_t^1$ are updated as in Step 11 of Algorithm 5.1, is termed as Algorithm 5.1_KCF-1. Similarly, the other special case of Algorithm 5.1 that uses only KCF-2 is termed as Algorithm 5.1_KCF-2, where its target appearance model $\hat{\mathbf{X}}_t^2$ and learned coefficients model $\hat{\boldsymbol{\alpha}}_t^2$ are updated as in Step 11 of Algorithm 5.1.

5.5 Experimental Results

Algorithm 5.1 is implemented in MATLAB and its average speed is 14.93 fps[1]. The algorithm when tested on gray scale/raw pixels is referred to as Algorithm 5.1_Gray,

[1]Using modern computer of 3.4GHz CPU and 16GB RAM

Algorithm 5.1 Tracking algorithm based on the kernelized correlation filters and the bilateral 2D principal component analysis subspace

Input: Target object is labeled in the first frame, and its initial state and location are $\mathbf{s}_1$ and (x_1, y_1), respectively.

Using $\mathbf{s}_1$ and (x_1, y_1), the target image $\mathbf{Y}_1$ is cropped out in the first frame and used to initialize the B2DPCA mean matrix $\boldsymbol{\mu}_{BP}$.

An image patch $\mathbf{X}_1$ is cropped out from (x_1, y_1) in the first frame and used to initialize both $\hat{\mathbf{X}}_1^1$ and $\hat{\mathbf{X}}_1^2$. Compute the coefficients $\boldsymbol{\alpha}_1^1$ using (5.2), and initialize $\hat{\boldsymbol{\alpha}}_1^1$ and $\hat{\boldsymbol{\alpha}}_1^2$ with $\boldsymbol{\alpha}_1^1$.

for $t > 1$ **do**

    An image patch $\mathbf{X}_t^D$ of new window size is cropped out from the position (x_{t-1}, y_{t-1}) in frame t, and then resized to match the initial window size.

    Compute $\overline{\mathbf{g}}_t^k$ from (5.3) using $\hat{\mathbf{X}}_{t-1}^k$, $\hat{\boldsymbol{\alpha}}_{t-1}^k$ and $\mathbf{X}_t^D$, where $k = 1, 2$.

    Compute $\widetilde{\mathbf{g}}_t$ using (5.9) after energy normalization of the response maps $\overline{\mathbf{g}}_t^k$, and the location of its maximum value is used to find (x_t, y_t).

    Sample M candidate states $\{\mathbf{s}_t^1, \mathbf{s}_t^2, ..., \mathbf{s}_t^M\}$ from $\mathbf{s}_{t-1}$ using the particle filter.

    Extract the candidate sample $\mathbf{Y}_t^m$ from the state $\mathbf{s}_t^m$ and position (x_t, y_t), $\forall m = 1, 2, ..., M$, and get centered candidate matrix $\overline{\mathbf{Y}}_t^m$.

    if $t <= T_{BP}$ **then**

        if $t <= 5$ **then**

            Assign $\mathbf{A}_t^m = \mathbf{J}$ (all-ones matrix) and $\mathbf{E} = \overline{\mathbf{Y}}_t^m$.

        else

            Assign $\mathbf{A}_t^m = \mathbf{J}$ and $\mathbf{E} = \overline{\mathbf{Y}}_t^m - \mathbf{U}_{BP}\,\mathbf{U}_{BP}^{\top}\,\overline{\mathbf{Y}}_t^m\,\mathbf{V}_{BP}\,\mathbf{V}_{BP}^{\top}$.

        end if

        Find d_{2DRC} for all $\mathbf{Y}_t^m$ using (5.11).

        Find the optimal state of the tracked target $\hat{\mathbf{s}}_t$ using (5.12) and (2.4).

        As described in section 5.4.3, update the observation models of KCF ($\hat{\mathbf{X}}_t^1$, $\hat{\boldsymbol{\alpha}}_t^1$, $\hat{\mathbf{X}}_t^2$, $\hat{\boldsymbol{\alpha}}_t^2$) incrementally every frame and that of B2DPCA ($\mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP}$) incrementally for every five frames by assigning $\mathbf{A}_t^m = \mathbf{J}$.

    else

        For all $\mathbf{Y}_t^m$, compute $\mathbf{Z}_t^m$ and $\mathbf{A}_t^m$ according to Procedure 5.1.

        Find the optimal state of the tracked target $\hat{\mathbf{s}}_t$ using (5.11), (5.12) and (2.4).

        The observation models of KCF ($\hat{\mathbf{X}}_t^1$, $\hat{\boldsymbol{\alpha}}_t^1$, $\hat{\mathbf{X}}_t^2$, $\hat{\boldsymbol{\alpha}}_t^2$) and B2DPCA ($\mathbf{U}_{BP}, \mathbf{V}_{BP}, \boldsymbol{\mu}_{BP}$) are updated incrementally for every one and five frames, respectively, as described in section 5.4.3.

    end if

end for

Output: Target state $\hat{\mathbf{s}}_t$ and position (x_t, y_t) at time t.

and when tested on histogram of oriented gradient (HOG) features is referred to as Algorithm 5.1_HOG. Their performance is evaluated on the challenging sequences available in OTB-50 [7] and VOT2016 [94] benchmark datasets using the respective evaluation protocols discussed in Section 2.3. The mapping function Φ and regularization parameter ξ in (5.1) of KCF-1 and KCF-2 are set to be as in KCF [71]. Even though the initialization of both KCFs, $(\hat{\mathbf{X}}_t^1,\ \hat{\boldsymbol{\alpha}}_t^1)$ and $(\hat{\mathbf{X}}_t^2,\ \hat{\boldsymbol{\alpha}}_t^2)$, is same as in Steps 1 and 2 of Algorithm 5.1, their observation model updates are different as in (5.14) and (5.16). In Algorithm 5.1, for B2DPCA representation, each image observation is resized to 32×32 pixels, and $k_l = 4$ left- and $k_r = 4$ right-projection basis vectors are used in all the experiments. The positive scalar ψ_{BP} used to terminate the IRRC technique in (5.8) is set to 0.1. The penalty constant λ_{BP} used in the computation of d_{2DRC} in (5.11) is set to 0.1. The B2DPCA observation model is incrementally updated for every 5 frames, and the occlusion ratio thresholds τ_1 and τ_2 used in B2DPCA observation model update are set to 0.1 and 0.6, respectively. The occlusion ratio threshold τ_{KCF} used in the model update of KCF-1 in (5.14) is set to 0.6. In both the correlation filters and B2DPCA, the cell size of 4×4 pixels with 9 orientations are adopted for extracting the HOG features.

The two parameters β_{BP} and δ_{BP} used in the weight function in (5.6) of the IRRC technique are computed as given below. In order to compute the value of β_{BP}, which controls the location of the demarcation point, the coding residuals $\hat{e}_{ij}$ at all locations are assumed to be in the "normal range" and follow the Gaussian distribution in the absence of occlusions in the tracked target candidate. But during occlusions, they will probably exceed the "normal range" at the occluded locations. Hence, by knowing the "normal range" of the residuals $\hat{e}_{ij}$ in the initial frames ($t = 2$ to T_{BP}) of the respective sequence, the value of β_{BP} is computed as

$$\beta_{BP} = \frac{1}{T_{BP} - 1} \sum_{t=2}^{T_{BP}} \left[\mathsf{mean}(\hat{\mathbf{E}}_t) + c_4\, \mathsf{std}(\hat{\mathbf{E}}_t) \right]^2 \tag{5.17}$$

where c_4 is a constant, $\hat{\mathbf{E}}_t$ is the residual error of the tracked target candidate, and T_{BP} is the time instant at which the number of the left- and right-projection basis vectors k_l and k_r become 4 for the first time. Note that the residual matrix $\hat{\mathbf{E}}_t$ is vectorized while computing the mean and standard deviation in (5.17). The first frame at $t = 1$ is manually labeled, due to which all the elements of the residual will be zero, and hence, the frame at $t = 1$ is not considered in the computation of β_{BP}. Now, the square of the residual $\hat{e}_{ij}$ that is larger than the computed β_{BP} will be considered as occluded/outlier and the value of weight a_{ij} will be less than 0.5. Further, the parameter δ_{BP}, which controls the rate of the weight between 1 and 0 is computed using $\delta_{BP} = c_5/\beta_{BP}$, where c_5 is a constant. The constants c_4 and c_5 are set as 2 and 7, respectively, for all the sequences. In Algorithm 5.1, the IRRC technique starts functioning after the number of the left- and right-projection basis vectors k_l and k_r become 4 for the first time. At time $t = T_{BP}$, the number of the left- and right-projection basis vectors k_l and k_r are 4, and then, the parameters β_{BP} and δ_{BP} are computed, which are then used by the IRRC technique for $t > T_{BP}$ to calculate the weights in (5.6).

The performance of the proposed Algorithm 5.1 is evaluated against several state-of-the-art tracking algorithms, namely, visual tracking via discriminative low-rank learning (DLR) [155], tracking via structured discriminative dictionary learning (DDL) [156], adaptive color attributes for real-time visual tracking (ACT) [67], visual tracking based on KCF [71], visual tracking via locally structured Gaussian process regression (LSGPR) [157], visual tracking based on discriminative subspace learning (DSL) [158] and the top two best tracking algorithms selected from the previous chapter based on their performance on each of the challenging attributes of both the OTB-50 and VOT2016 benchmark datasets, for a fair comparison. The OTB-50 results of the trackers DDL [156], DLR [155], LSGPR [157] and DSL [158] are available on the respective authors' website [159–161] and [162], and they have been used to compare the performance with that of Algorithm 5.1 on OTB-50 dataset. The codes of the

trackers ACT [67] and KCF [71] downloaded from the respective authors' website [163] and [164] are used to evaluate on the sequences of the two benchmarks to have a fair comparison with Algorithm 5.1. The parameter settings of these trackers are as given in their respective papers in all the experiments. Note that visual tracking based on KCF [71] is evaluated on gray values and HOG features, which are denoted as KCF_Gray and KCF_HOG, respectively, to compare the performance with that of Algorithm 5.1. In Algorithm 5.1, 400 particles are used to find the four affine parameters of the target $(\theta_t, s_t, \alpha_t, \phi_t)$ using B2DPCA and 2DRC. In the following sub-sections, two benchmark datasets and the evaluation measures discussed in Section 2.3 are used for the quantitative and qualitative evaluation of Algorithm 5.1 with that of the other methods.

5.5.1 Performance Evaluation on OTB-50

a) Special cases of Algorithm 5.1

Two special cases of Algorithm 5.1 are considered each with only one KCF enabled and other disabled to analyze their effect on the tracking performance. Compared to Algorithm 5.1, the methods in Algorithm 5.1_KCF-1 and Algorithm 5.1_KCF-2, use only one correlation filter, KCF-1 and KCF-2, respectively, to find the location of the target (x_t, y_t). The performance of special cases of Algorithm 5.1 are compared with that of Algorithm 5.1 in terms of the *precision score* and area under curve (AUC) on OTB-50 benchmark dataset in Table 5.1. For this comparison, Algorithm 5.1 and the special cases of Algorithm 5.1 are tested on HOG features. It is observed from the Table 5.1 that Algorithm 5.1_KCF-1 performs better than Algorithm 5.1_KCF-2 does in terms of the *precision score* and AUC. Also, notice that the combination of the two correlations filters, KCF-1 and KCF-2, further enhances the performance of Algorithm 5.1 in terms of both the *precision score* and AUC.

Table 5.1: Performance comparison of special cases of Algorithm 5.1 in terms of the *precision score* and AUC on OTB-50.

Variants/Metric	***Precision Score***	**AUC**
Algorithm 5.1_KCF-1	75.8	55.0
Algorithm 5.1_KCF-2	73.2	53.2
Algorithm 5.1	77.7	56.2

b) Comparison with the state-of-the-art algorithms:

Algorithm 5.1 is evaluated on the OTB-50 benchmark dataset [7] and compared with the state-of-the-art tracking algorithms using one-pass evaluation (OPE). Table 5.2 shows the performance comparison of Algorithm 5.1 in terms of the *precision score* for the location error threshold of 20 pixels with that of the other state-of-the-art trackers for different attributes. The best three results are shown in **(red, bold)**, (violet, underline) and *(blue, italic)* fonts for better comparison of Algorithm 5.1 with the other methods. It can be observed from Table 5.2 that Algorithm 5.1_HOG outperforms the other trackers in all the challenging attributes, whereas Algorithm 4.1 stood third in low resolution (LR). KCF_HOG ranks first in background clutter (BC), and second in the remaining attributes except scale variation (SV) and LR. Further, DLR stands second in SV and third in the remaining attributes except deformation (Def), motion blur (MB), fast motion (FM), in-plane rotation (IPR) and LR. DDL ranks second in BC and third in out-of-plane rotation (OPR), SV, Def and IPR. Also, ACT stands third in MB and FM, whereas LSST stand second in LR. It is also observed from the Table 5.2 that Algorithm 5.1_Gray, based on gray features, outperforms Algorithm 3.1_HOG, Algorithm 4.1 and other particle filter-based trackers (LDSAM, ASLA, L1APG, LSST, LWIST and WLCS) for MB and FM challenging attributes. This improvement in performance is due to the collaboration of KCF and B2DPCA used in Algorithm 5.1. Further, Algorithm 5.1_HOG outperforms KCF_HOG in all the attributes especially in SV, FM, out-of-view (OV) and LR attributes by 10.3%, 10.8%, 12.3% and 22.6%, respectively. It is also observed

that Algorithm 5.1_HOG has a performance superior to that of Algorithm 3.1_HOG, Algorithm 4.1 and other methods for all the challenging attributes.

The performance comparison of Algorithm 5.1 in terms of AUC with that of the other state-of-the-art trackers for different attributes is shown in Table 5.3. It is observed from the table that the performance of Algorithm 5.1_HOG is superior to that of the other methods in all the challenging attributes except Def, whereas Algorithm 4.1 stood second in LR. Further, KCF_HOG ranks first in Def and second in the remaining attributes except illumination variation (IV), OPR, SV and LR, whereas DDL ranks third in SV and BC, and second in the remaining challenges except MB, FM, OV and LR. Further, LSST stands third in LR, and ACT and DSL stands third in MB and IPR, respectively. Also, DLR ranks second in SV and IPR, and third in the remaining attributes except Def, MB, BC and LR. It is observed from Table 5.3 that the performance of Algorithm 5.1_Gray is superior to that of Algorithm 3.1_HOG, Algorithm 4.1 and other particle filter-based trackers (LDSAM, ASLA, L1APG, LSST, LWIST and WLCS) for MB and FM challenging attributes. Also, it is observed that Algorithm 5.1_HOG outperforms KCF_HOG in all the attributes especially in IV, OPR, SV, FM, OV and LR attributes by 11.9%, 9.3%, 27.4%, 9.6%, 9.4% and 90.4%, respectively. Further, Algorithm 5.1_HOG outperforms Algorithm 3.1_HOG, Algorithm 4.1 and other methods for all the challenging attributes except in Def.

The performance of Algorithm 5.1 is compared using the *precision* and *success plots* of OPE for OTB-50 benchmark sequences having occlusion against the other trackers and shown in Figure 5.3. In the *precision plot*, the *precision score* for the location error threshold of 20 pixels is used to rank the tracker, whereas in the *success plot*, AUC is used to rank the overall performance of the tracker. It is observed that the performance of Algorithm 5.1_HOG is superior to that of the state-of-the-trackers KCF_HOG, DLR, DDL, DSL, LSGPR, ACT, Algorithm 4.1, LDSAM, DLT, Algorithm 3.1_HOG, ASLA, KCF_Gray and L1APG by 6.0%, 13.6%, 15.4%, 21.2%,

Table 5.2: The *precision score* of Algorithm 5.1 with that of the compared trackers for different attributes of OTB-50. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Precision score	**IV**	**OPR**	**SV**	**Occ**	**Def**	**MB**	**FM**	**IPR**	**OV**	**BC**	**LR**
Algorithm 5.1_HOG	**74.8**	**76.0**	**74.9**	**79.4**	**74.7**	**68.6**	**66.7**	**73.2**	**73.0**	**75.3**	**77.1**
Algorithm 5.1_Gray	42.7	51.4	51.4	50.8	48.3	45.4	41.9	49.9	35.2	48.1	57.0
Algorithm 4.1	61.5	62.9	63.4	60.4	59.8	33.9	33.3	58.6	39.9	60.1	*68.1*
Algorithm 3.1_HOG	56.5	56.3	60.3	55.8	52.6	36.5	35.1	58.3	51.0	57.4	55.7
KCF_HOG	72.8	72.9	67.9	74.9	74.0	65.0	60.2	72.5	65.0	**75.3**	62.9
KCF_Gray	44.8	54.1	49.2	50.5	48.0	39.4	44.1	55.2	35.8	50.3	52.9
ACT	57.5	64.5	59.8	61.9	60.5	*55.0*	*48.0*	67.5	43.4	62.9	55.9
LSGPR	52.1	57.1	58.6	62.7	57.9	31.2	33.6	53.5	45.5	57.0	55.1
DSL	62.7	69.1	68.6	65.5	67.6	37.2	41.1	63.8	41.3	62.0	67.6
DLT	53.4	56.1	59.0	57.4	56.3	45.3	44.6	54.8	44.4	49.5	53.6
DDL	65.0	*72.6*	*69.3*	68.8	*69.5*	33.2	36.2	*67.9*	35.4	67.3	62.8
DLR	*65.9*	*72.6*	71.7	*69.9*	69.1	42.0	45.8	67.2	*51.1*	*66.2*	67.6
LDSAM	58.8	61.6	60.1	58.1	59.9	29.9	28.4	59.4	45.9	62.4	61.2
ASLA	56.4	59.1	62.3	53.3	57.7	32.3	33.1	57.5	41.7	63.6	59.0
L1APG	34.1	47.8	47.2	46.1	38.3	37.5	36.5	51.8	32.9	42.5	61.5
LSST	41.5	47.0	53.3	41.3	45.0	27.1	25.2	46.1	24.3	42.6	74.1
LWIST	34.3	38.4	40.3	40.4	35.3	15.1	18.7	33.7	30.7	37.4	45.9
WLCS	32.9	33.0	38.3	35.7	29.9	15.4	21.3	30.3	22.8	33.1	23.0

Table 5.3: AUC of Algorithm 5.1 with that of the compared trackers for different attributes of OTB-50. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

AUC	IV	OPR	SV	Occ	Def	MB	FM	IPR	OV	BC	LR
Algorithm 5.1_HOG	**55.2**	**54.1**	**54.4**	**55.9**	*52.2*	**51.3**	**50.3**	**53.3**	**60.2**	**54.3**	**51.8**
Algorithm 5.1_Gray	32.0	35.1	33.6	35.7	34.3	35.7	32.7	35.4	32.1	35.3	28.3
Algorithm 4.1	49.3	47.8	49.4	46.6	46.2	28.7	28.6	45.1	35.2	47.1	47.2
Algorithm 3.1_HOG	46.6	44.6	48.2	44.0	42.0	31.6	30.2	46.4	43.2	46.4	41.6
KCF_HOG	49.3	49.5	42.7	51.4	**53.4**	49.7	45.9	49.7	55.0	53.5	27.2
KCF_Gray	34.8	38.7	34.8	37.3	36.4	34.5	37.1	39.2	35.3	37.2	25.8
ACT	41.4	44.1	38.4	42.5	43.4	*41.0*	37.3	46.9	41.0	44.9	26.7
LSGPR	41.9	42.3	43.7	48.1	45.4	28.0	28.8	39.2	40.8	43.7	33.0
DSL	48.0	50.2	51.5	48.1	49.9	33.1	36.0	*47.1*	36.3	46.4	38.5
DLT	40.5	41.2	45.5	42.3	39.4	36.3	36.0	41.1	36.7	33.9	34.7
DDL	51.1	53.7	*52.6*	51.4	53.3	31.6	33.4	49.7	34.4	*51.0*	39.0
DLR	*50.9*	*52.7*	54.1	*51.2*	51.5	36.4	*39.0*	49.7	*44.2*	49.8	38.5
LDSAM	47.9	47.5	47.5	45.9	48.0	27.3	26.5	46.3	40.4	49.4	44.1
ASLA	46.6	46.5	49.0	43.1	46.2	28.2	29.2	45.6	38.7	50.2	42.1
L1APG	28.3	36.0	35.0	35.3	31.1	31.0	31.1	39.1	30.3	35.0	37.4
LSST	33.2	33.5	37.7	30.4	31.6	21.4	21.5	32.3	21.7	30.2	*46.2*
LWIST	28.9	30.1	31.3	31.9	28.5	15.1	18.3	27.2	28.1	31.1	27.8
WLCS	27.0	25.8	29.5	27.6	25.0	15.8	19.6	24.2	21.6	27.3	13.7

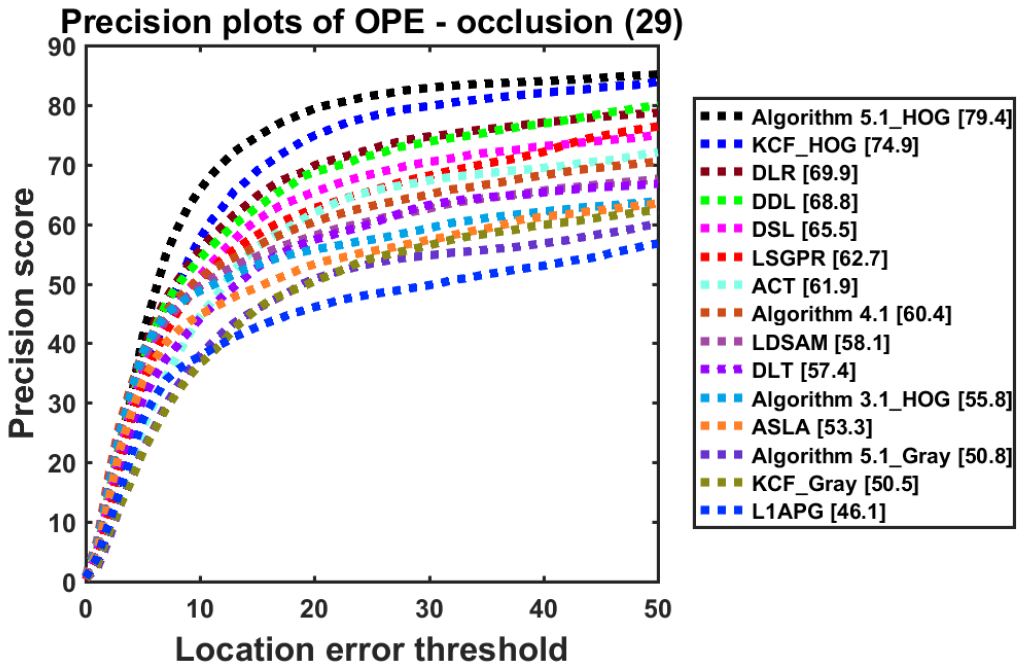
26.6%, 28.3%, 31.4%, 36.6%, 38.3%, 42.3%, 48.9%, 57.2% and 72.2%, respectively, in terms of the *precision score*. On the other hand, Algorithm 5.1_HOG outperforms DDL, KCF_HOG, DLR, DSL, LSGPR, Algorithm 4.1, LDSAM, Algorithm 3.1_HOG, ASLA, ACT, DLT, KCF_Gray and L1APG by 8.7%, 8.7%, 9.2%, 16.2%, 16.2%, 19.9%, 21.8%, 27%, 29.7%, 31.5%, 32.1%, 49.8% and 58.3%, respectively, in terms of AUC. Further, it is observed that Algorithm 5.1_HOG outperforms both DDL and KCF_HOG in terms of the *precision score* and the *success rate* for the location error threshold > 5 and overlap threshold < 0.6, respectively. This is due to the outliers/occlusion handling capability of Algorithm 5.1 in both the observation model update and the computation of the observation likelihood in the generative model. For sequences having occlusion, it is observed that Algorithm 5.1_HOG yields a performance superior to that of Algorithm 3.1_HOG, Algorithm 4.1 and other trackers in terms of both the *precision score* and the *success rate*.

The *precision* and *success plots* of OPE for the various trackers averaging over the OTB-50 benchmark sequences are shown in Figure 5.4. To rank the tracker, the *precision score* for the threshold of 20 pixels is used in the *precision plot*, whereas AUC is used in the *success plot*, and their values are shown along with the tracker name. From Figure 5.4, it is observed that Algorithm 5.1_HOG outperforms the state-of-the-trackers KCF_HOG, DLR, DDL, DSL, Algorithm 4.1, ACT, LDSAM, LSGPR, ASLA, Algorithm 3.1_HOG, DLT, KCF_Gray and LSST by 5.0%, 8.6%, 9.1% 13.7%, 22.5%, 23.5%, 23.7%, 26.3%, 27.1%, 27.4%, 32.3%, 38.7% and 57.3%, respectively, in terms of the *precision score*. Similarly in terms of the AUC, Algorithm 5.1_HOG outperforms DDL, DLR, KCF_HOG, DSL, LDSAM, Algorithm 4.1, ASLA, Algorithm 3.1_HOG, LSGPR, ACT, DLT, KCF_Gray and L1APG, by 4.2%, 5.0%, 9.3%, 10.2%, 13.5%, 13.7%, 15.1%, 15.9%, 21.1%, 26.8%, 28.9%, 38.4% and 47.9%, respectively. It is also observed that Algorithm 5.1_HOG outperforms both DDL and KCF_HOG in terms of the *precision score* and the *success rate* for the location error threshold > 5 and overlap threshold < 0.6, respectively. Overall, Algorithm 5.1_HOG provides the

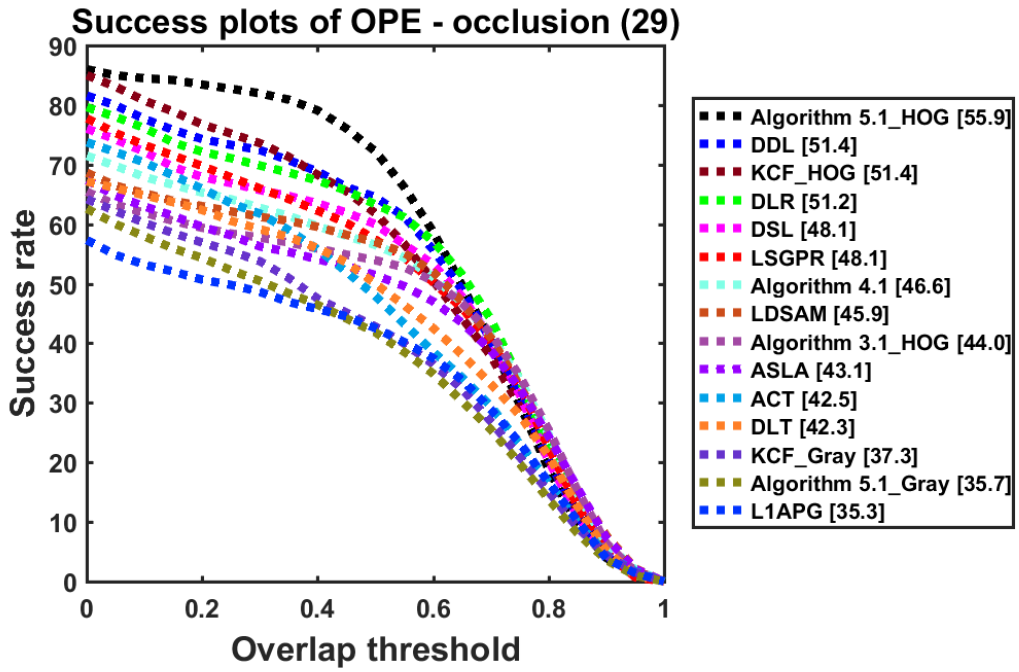
best performance to that of Algorithm 3.1_HOG, Algorithm 4.1 and other trackers in terms of both the *precision score* and the *success rate*.

5.5.2 Performance Evaluation on VOT2016

The performance of Algorithm 5.1 is evaluated on the VOT2016 benchmark dataset [94] using the average accuracy and robustness. The accuracy rank and overlap comparison of Algorithm 5.1 with that of the recent state-of-the-art tracking algorithms averaging over the VOT2016 sequences is shown in Table 5.4. Likewise, Table 5.5 shows the robustness rank and failures comparison of Algorithm 5.1 averaging over the same challenging sequences. Also, the respective measures with different averaging methodologies, mean, weighted mean and pooled, are shown in the last six columns of these two tables. For a better comparison of Algorithm 5.1 with that of the other state-of-the-art trackers, the best three results are shown in **(red, bold)**, (violet, underline) and *(blue, italic)* fonts. From Table 5.4, it is observed that in terms of the overlap, Algorithm 5.1_HOG ranks first for the attributes mean, weighted mean and pooled, and stands second for the attributes camera motion, motion change and size change, and third for the attributes empty, illumination change and occlusion. Also, Algorithm 4.1 ranks second in attribute empty and third in attributes size change and mean, whereas Algorithm 3.1_HOG ranks first in camera motion and occlusion, second in mean, weighted mean and pooled attributes. Further, in terms of the overlap, KCF_HOG stands first for the attributes camera motion and occlusion, third for the attributes motion change, weighted mean and pooled. Also, ACT stands first, second and third for the attributes motion change, occlusion and camera motion, respectively. Further, LWIST and WLCS ranks first and second for the attribute illumination change, respectively. Algorithm 5.1_HOG has a performance better than that of KCF_HOG in the face of all the challenging attributes except camera motion and occlusion. It is also noticed that Algorithm 5.1_HOG performs better than Algorithm 3.1_HOG does for all the challenging attributes except empty

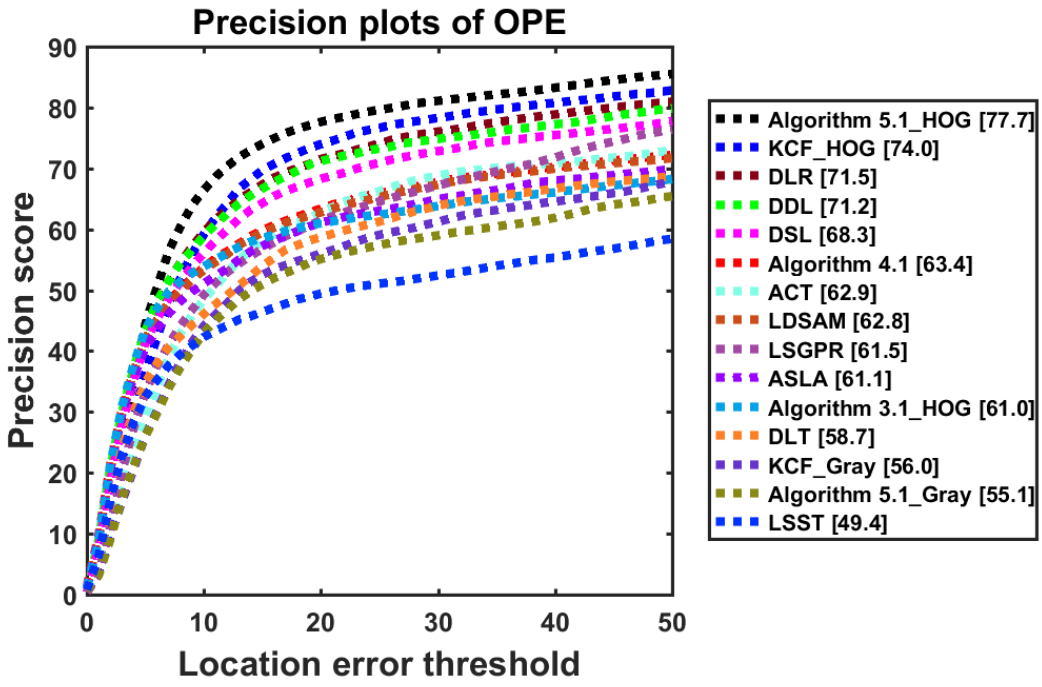


(a)

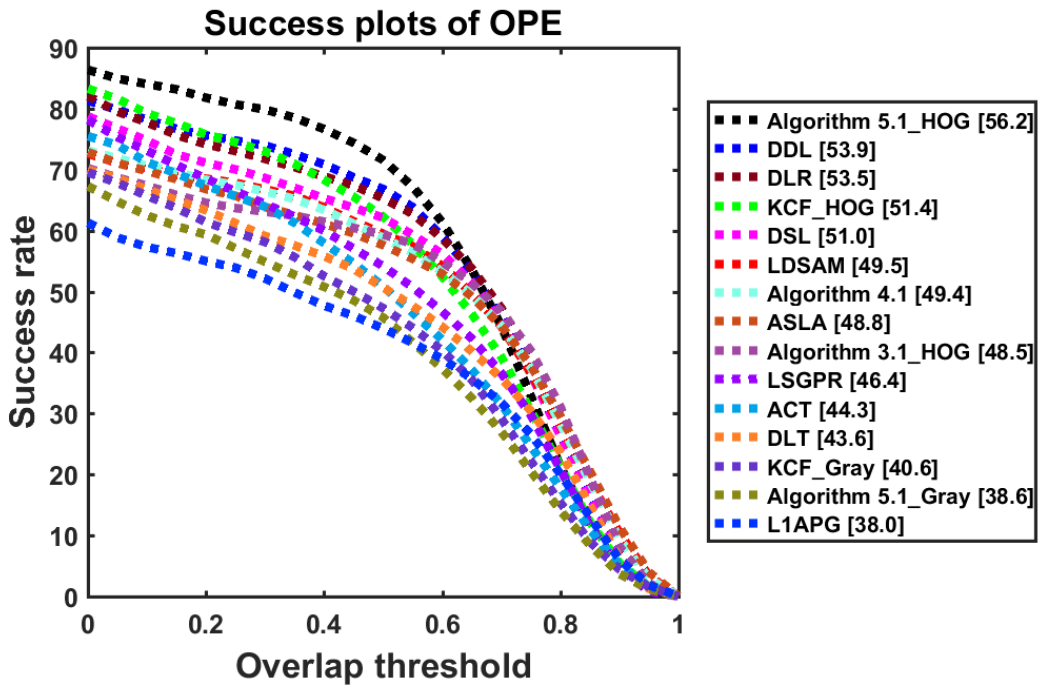


(b)

Figure 5.3: Performance evaluation of Algorithm 5.1 on OTB-50 sequences having occlusion using (a) the *precision plots* of OPE, where the *precision score* is shown along with the tracker name in the legend, and (b) the *success plots* of OPE, where AUC is shown along with the tracker name in the legend. The number 29 appearing in the title denotes the number of sequences associated with the occlusion attribute of OTB-50 dataset.



(a)



(b)

Figure 5.4: Overall performance evaluation of Algorithm 5.1 on OTB-50 using (a) the *precision plots* of OPE, where the *precision score* is shown along with the tracker name in the legend, and (b) the *success plots* of OPE, where AUC is shown along with the tracker name in the legend.

and size change. Further, Algorithm 5.1_HOG outperforms Algorithm 4.1 for all the challenging attributes except empty.

From Table 5.5, it is observed that Algorithm 5.1_HOG ranks second in the attributes camera motion and empty, and third in size change, mean, weighted mean and pooled, whereas Algorithm 5.1_Gray stand third in the attributes motion change and occlusion. On the other hand, KCF_HOG stands second in the attributes motion change, size change, mean, weighted mean and pooled, and third in camera motion, whereas KCF_Gray ranks first in empty, and second in illumination change, motion change, occlusion and mean. Further, ACT ranks first in all the attributes in terms of failures except empty, where it stood third, whereas ASLA stood third in illumination change. Algorithm 5.1_HOG has a performance better or similar to that of KCF_HOG in the face of all the challenging attributes except motion change, occlusion, size change, mean, weighted mean and pooled. This may happen when the target undergoes severe occlusion, scale change, out-of-plane rotation, motion blur, fast motion either individually or simultaneously. It is also noticed that Algorithm 5.1_HOG has a performance better than that of Algorithm 3.1_HOG and Algorithm 4.1 for all the challenging attributes except illumination change. Further, the overall performance (mean, weighted mean and pooled attributes) of Algorithm 5.1_HOG is superior to that of the other methods in terms of overlap, but its performance is not that good in terms of failures, where it stood third.

5.5.3 Qualitative Evaluation

For qualitative evaluation of the trackers, some tracking results on a subset of the OTB-50 benchmark sequences are obtained and shown in Figure 5.5. The tracking results of top twelve trackers, which are selected from Figure 5.4, on the six exemplar image frames are shown for each sequence and these six frames are selected at regular intervals without any bias. Algorithm 5.1_HOG successfully tracks the target in all the frames of the *Car4*, *CarScale*, *Fleetface*, *Freeman3*, *Freeman4*, *Jogging-1*, *Singer1*,

Table 5.4: Accuracy rank (A-Rank) and average overlap comparison of Algorithm 5.1 with that of the compared trackers for different attributes of VOT2016. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Accuracy	camera motion		empty		illum. change		motion change		occlusion		size change		Mean		Weighted mean		Pooled	
Trackers	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap
Algorithm 5.1_HOG	**1.00**	0.48	**1.00**	*0.51*	**1.00**	*0.62*	**1.00**	0.42	**1.00**	*0.40*	**1.00**	0.42	**1.00**	**0.47**	**1.00**	**0.46**	**1.00**	**0.47**
Algorithm 5.1_Gray	**1.00**	0.43	**1.00**	0.50	*9.00*	0.43	**1.00**	0.39	2.00	0.34	8.00	0.32	3.67	0.40	3.67	0.41	**1.00**	0.43
Algorithm 4.1	**1.00**	0.45	**1.00**	0.52	**1.00**	0.61	**1.00**	0.39	**1.00**	0.36	**1.00**	*0.42*	**1.00**	*0.46*	**1.00**	0.45	**1.00**	0.46
Algorithm 3.1_HOG	**1.00**	0.45	**1.00**	**0.52**	**1.00**	0.61	**1.00**	0.38	**1.00**	0.38	**1.00**	**0.43**	**1.00**	0.46	**1.00**	0.45	**1.00**	0.46
KCF_HOG	**1.00**	**0.50**	**1.00**	0.49	*9.00*	0.44	**1.00**	*0.41*	**1.00**	**0.43**	3.00	0.36	2.67	0.44	2.67	*0.45*	**1.00**	*0.46*
KCF_Gray	**1.00**	0.42	**1.00**	0.49	*9.00*	0.44	**1.00**	0.40	2.00	0.35	8.00	0.31	3.67	0.40	3.67	0.41	**1.00**	0.43
ACT	2.00	*0.46*	**1.00**	0.50	*9.00*	0.44	**1.00**	**0.43**	**1.00**	0.41	*6.00*	0.35	3.33	0.43	3.33	0.44	**1.00**	0.45
WLCS	*3.00*	0.41	**1.00**	0.50	**1.00**	0.63	5.00	0.34	2.00	0.33	**1.00**	0.40	2.17	0.43	2.17	0.42	**1.00**	0.43
LWIST	*3.00*	0.42	**1.00**	0.49	**1.00**	**0.65**	5.00	0.34	2.00	0.34	**1.00**	0.41	2.17	0.44	2.17	0.42	**1.00**	0.43
DLT	**1.00**	0.44	**1.00**	0.50	**1.00**	0.58	**1.00**	0.39	**1.00**	0.38	**1.00**	0.42	**1.00**	0.45	**1.00**	0.44	**1.00**	0.45
LSST	*3.00*	0.40	**1.00**	0.44	2.00	0.55	5.00	0.32	2.00	0.34	**1.00**	0.37	*2.33*	0.40	*2.33*	0.39	2.00	0.40
LDSAM	11.00	0.33	14.00	0.36	14.00	0.31	*11.00*	0.28	*4.00*	0.31	11.00	0.25	10.83	0.31	10.83	0.31	*13.00*	0.33
ASLA	**1.00**	0.44	**1.00**	0.49	**1.00**	0.61	**1.00**	0.38	**1.00**	0.36	**1.00**	0.41	**1.00**	0.45	**1.00**	0.44	**1.00**	0.44
L1APG	**1.00**	0.43	**1.00**	0.50	*9.00*	0.45	**1.00**	0.37	2.00	0.35	*6.00*	0.33	3.33	0.40	3.33	0.41	**1.00**	0.42

Table 5.5: Robustness rank (R-Rank) and average failures comparison of Algorithm 5.1 with that of the compared trackers for different attributes of VOT2016. **(red, bold)**, (violet, underline) and *(blue, italic)* indicate first, second and third rankings, respectively.

Robustness	camera motion		empty		illum. change		motion change		occlusion		size change		Mean		Weighted mean		Pooled	
Trackers	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures
Algorithm 5.1_HOG	**1.00**	66.93	**1.00**	35.07	*3.00*	10.07	2.00	64.13	**1.00**	25.47	**1.00**	*33.33*	*1.50*	*39.17*	*1.50*	*47.29*	2.00	*158.60*
Algorithm 5.1_Gray	3.00	73.87	**1.00**	36.27	2.00	9.33	**1.00**	*57.80*	**1.00**	*24.93*	*5.00*	41.20	2.17	40.57	2.17	49.79	2.00	167.53
Algorithm 4.1	*4.00*	90.87	2.00	40.53	*3.00*	9.13	2.00	74.40	**1.00**	28.27	**1.00**	33.73	2.17	46.16	2.17	57.59	*6.00*	192.80
Algorithm 3.1_HOG	7.00	109.27	10.00	62.53	*3.00*	10.00	*8.00*	91.40	2.00	36.20	*5.00*	42.73	5.83	58.69	5.83	73.54	9.00	248.00
KCF_HOG	**1.00**	*68.00*	2.00	39.00	*3.00*	10.00	**1.00**	57.00	**1.00**	25.00	**1.00**	32.00	*1.50*	38.50	*1.50*	47.08	2.00	157.00
KCF_Gray	**1.00**	72.00	**1.00**	**35.00**	**1.00**	7.00	**1.00**	57.00	**1.00**	24.00	2.00	36.00	1.17	38.50	1.17	47.74	**1.00**	160.00
ACT	**1.00**	**62.93**	**1.00**	*36.00*	**1.00**	**5.00**	**1.00**	**47.00**	**1.00**	**23.00**	**1.00**	**26.93**	**1.00**	**33.48**	**1.00**	**41.93**	**1.00**	**140.00**
WLCS	11.00	164.73	13.00	89.53	*3.00*	9.13	11.00	124.13	11.00	46.67	11.00	60.87	10.00	82.51	10.00	105.86	13.00	355.27
LWIST	10.00	131.27	11.00	81.13	*3.00*	8.53	10.00	115.13	10.00	43.07	10.00	55.27	9.00	72.40	9.00	91.28	11.00	307.53
DLT	*4.00*	94.73	**1.00**	42.20	*3.00*	10.07	2.00	66.53	2.00	35.40	*5.00*	43.33	2.83	48.71	2.83	60.06	*6.00*	201.47
LSST	10.00	135.60	11.00	77.87	5.00	11.40	10.00	115.20	10.00	45.60	12.00	61.47	9.67	74.52	9.67	93.11	11.00	315.20
LDSAM	13.00	167.13	13.00	114.20	*3.00*	10.00	13.00	131.40	10.00	51.67	14.00	76.60	11.00	91.83	11.00	117.10	13.00	396.53
ASLA	*4.00*	92.80	*5.00*	45.80	2.00	*7.73*	*8.00*	79.27	2.00	33.93	*5.00*	40.20	4.33	49.96	4.33	61.84	7.00	208.27
L1APG	10.00	140.80	*5.00*	49.13	5.00	11.60	10.00	99.13	*7.00*	38.60	10.00	54.27	7.83	65.59	7.83	82.89	9.00	267.27

Suv and *Trellis* sequences, which contain most of the real-time challenges such as pose change, partial occlusion, illumination change, scale change and out-of-plane rotation. This indicates the strong capabilities of Algorithm 5.1 in handling these challenges. Even though Algorithm 5.1_HOG performs well for the challenges that are considerably difficult, it loses the target for challenges that are extremely difficult or for the sequences where the target undergoes several changes simultaneously. This can be observed in the last image (#1332) of the *Lemming* sequence, one of the longest and most challenging sequences, where the target undergoes severe occlusion, scale change, out-of-plane rotation, motion blur, fast motion either individually or simultaneously. Algorithm 5.1_HOG has tracked the target successfully till frame #1110 without loosing in between and has failed afterwards (towards the end of the sequence). In contrast to Algorithm 5.1_HOG, other methods have failed in the middle of the *Lemming* sequence, but at different frames of the sequence. For example, in frame #0444 and #0666, all the trackers have failed except Algorithm 5.1_HOG and DLT, whereas in #0888, DLT has failed, and all the remaining trackers have started to track the target again except Algorithm 3.1_HOG, Algorithm 4.1, LDSAM, ASLA, DDL and LSGPR. So, none of the trackers has tracked the target through the entire *Lemming* sequence successfully. Even though some trackers fail in some frames, they are able to track the object once again by chance as the object reappears at the same location due to camera pan or due to repetitive motion of the object. All the trackers perform well in the *Car4* sequence except ACT, where the tracker drifts away slightly but with imprecise estimation of scale. In the *CarScale* sequence, all the methods track the target, but fail to estimate the scale and location of the target effectively. Similar observations can be made even in the *Fleetface* sequence, where ACT, ASLA, Algorithm 3.1_HOG and Algorithm 4.1 fail towards the end of the sequence, and LDSAM fails to estimate the scale of the object accurately. Algorithm 5.1_HOG, Algorithm 3.1_HOG, Algorithm 4.1, LDSAM and DLT estimate both the scale and location of the target effectively in *Freeman3*, whereas DSL and DLR fail to estimate

the scale of the target. The only method that tracks the target effectively until the end in the *Freeman4* sequence is Algorithm 5.1_HOG, and Algorithm 4.1 has drifted away little bit towards the end of the sequence. Further, in the *Jogging-1* sequence, LSGPR, DDL, ASLA and Algorithm 5.1_HOG track the object successfully, whereas the other trackers fail to track the object after a few initial frames. In *Singer1*, all the methods track the target to successfully except DLR and ACT fail to estimate the scale accurately. Also, Algorithm 5.1_HOG, Algorithm 3.1_HOG, KCF_HOG and LSGPR track the object successfully in *Suv* except DDL, DLT, DLR, DSL, ACT, Algorithm 4.1, LDSAM and ASLA, which fail to track the object towards the end of the sequence. Further, Algorithm 5.1_HOG, Algorithm 3.1_HOG and ASLA successfully track the target in *Trellis*. Eventhough KCF_HOG, DLR, DDL and DSL track the target in *Trellis*, but they fail to estimate the scale precisely. Also, ACT fails to track the target in the middle of the *Trellis* sequence (#470) and starts to track again but with imprecise scale and location (#564). Further, Algorithm 4.1, LDSAM, LSGPR and DLT fail to track the object towards the end of the sequence in *Trellis*. Thus, from these qualitative analyses, it is observed that Algorithm 5.1_HOG performs favorably in most of the challenging sequences and better than that of Algorithm 3.1, Algorithm 4.1 and other methods.

5.6 Summary

In this chapter, a new tracking algorithm, Algorithm 5.1, which is based on a collaboration of the discriminative and generative models has been proposed. In the discriminative model, kernelized correlation filters have been used to find the target position, and a new generative model has been used to find the remaining affine motion parameters of the target. In the generative model, the robust coding technique used in the first tracking algorithm has been extended to two dimensions, and then, used in the reconstruction procedure of bilateral 2DPCA to develop an iteratively

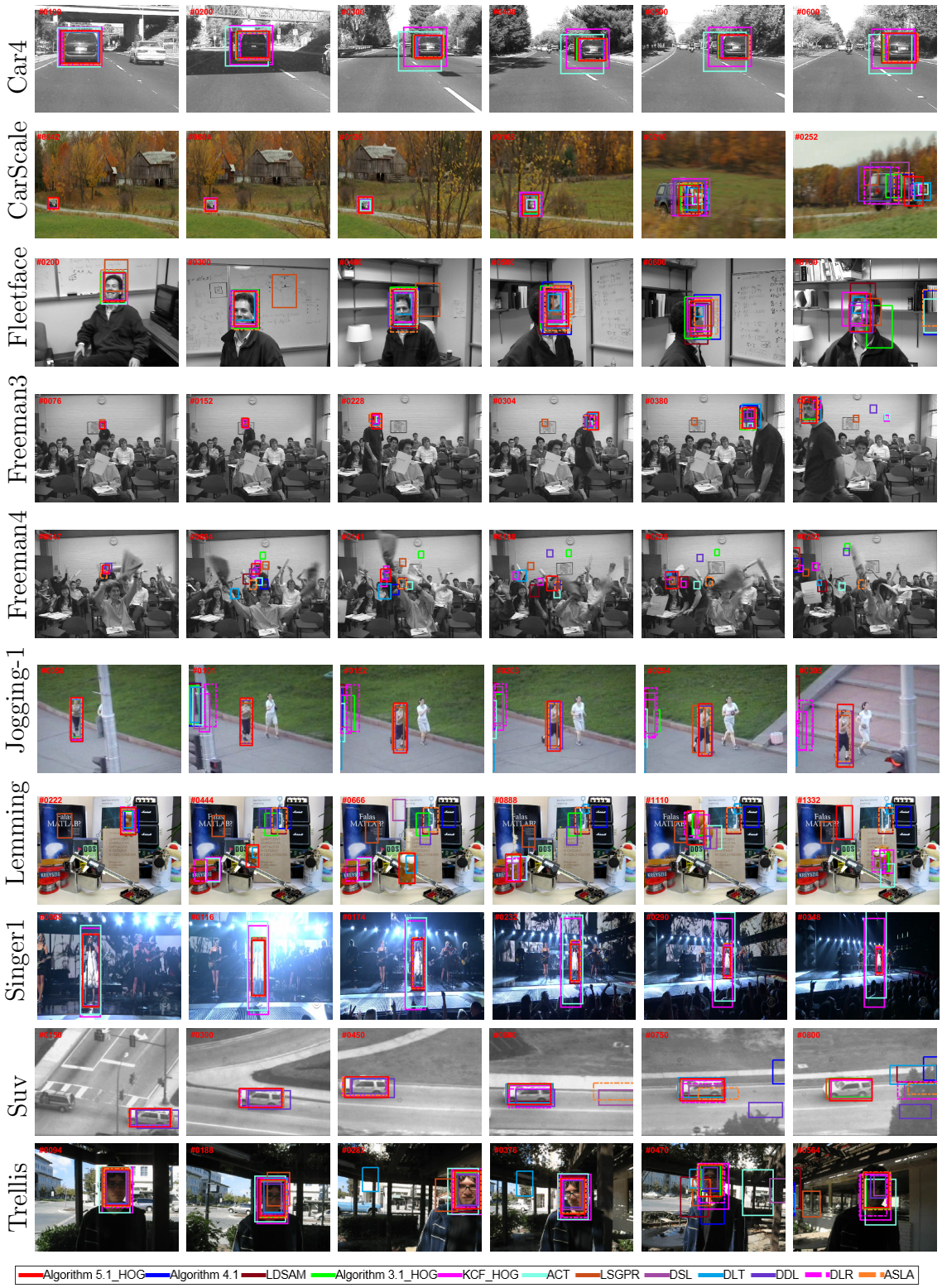


Figure 5.5: Examples of tracking results of the compared methods on the ten OTB-50 benchmark sequences.

reweighted robust coding technique. A 2D robust coding distance measure has been defined, and then used to compute the observation likelihood. By exploiting the weights obtained from iteratively reweighted robust coding, a robust occlusion map has been generated and used in the observation model update of both the kernelized correlation filters and the bilateral 2DPCA subspace. Extensive experiments have been conducted on the two popular benchmark datasets, OTB-50 and VOT2016, to analyze the performance of Algorithm 5.1. Performance of Algorithm 5.1 has been compared with that of Algorithm 3.1, Algorithm 4.1 and the other methods using these datasets. Algorithm 5.1 generally outperforms these methods for most of the challenging attributes of OTB-50, both in terms of the *precision score* and area under curve (AUC). The performance of Algorithm 5.1_HOG is almost comparable to that of ACT in terms of failures for all the challenging attributes of the VOT2016 including mean, weighted mean and pooled attributes. On the other hand, even though none of the trackers perform well for all the challenging attributes of the VOT2016 in terms of overlap, Algorithm 5.1_HOG outperforms all the other methods when the overlap measure is averaged over all the challenging attributes (mean, weighted mean and pooled). Quantitative and qualitative performance of Algorithm 5.1_HOG have shown that Algorithm 5.1_HOG generally outperforms the other methods for most of the challenging attributes of both OTB-50 and VOT2016 datasets as well as when averaged over these attributes. In the next chapter, the concluding remarks and some suggestions for the future investigation are presented.

Chapter 6

Conclusion

6.1 Concluding Remarks

Visual tracking has drawn a great deal of attention among the research community since the last decade due to its wide range of real-life applications in the field of computer vision, such as action recognition, vehicle navigation, robotics, human behavior analysis, human computer interaction, event/activity detection, sports video analysis, video indexing and retrieval, medical imaging, traffic management, security, and surveillance. In this thesis, three visual object tracking algorithms have been developed to improve the tracking performance over that of the existing algorithms by exploring different object representation schemes to model the object appearance, and by devising mechanisms to update the observation models.

A new tracking algorithm (Algorithm 3.1) based on the global appearance model using robust coding and its collaboration with a local model has been proposed. The global appearance of the object has been modeled by the global PCA subspace, and an iteratively reweighted robust coding (IRRC) technique has been developed to compute the optimum global PCA basis coefficients and the global weight matrix. The global weight matrix and the local PCA model have been used to find the optimum local PCA basis coefficients, and these models are collaborated to exploit their individual advantages. Global and local robust coding distances have been introduced to

measure the similarity between the candidate sample and the corresponding sample reconstructed from the subspace by reducing the effects of outliers/occlusion on these distances. A novel observation likelihood based on both the global and local robust coding distances has been introduced, and then used to obtain the tracking result. The global weights obtained during IRRC have been exploited to detect the outliers/occlusions as well as to generate the occlusion map, which has been used to update both the global and local observation models. Extensive experiments have been conducted on the two popular benchmark datasets, the object tracking benchmark-50 (OTB-50) [7] and the visual object tracking 2016 (VOT2016) [94], to evaluate Algorithm 3.1. The algorithm has been tested on the gray scale features (Algorithm 3.1_Gray) and histogram of oriented gradient (HOG) features (Algorithm 3.1_HOG) for its performance comparison with that of the several state-of-the-art algorithms in the framework of particle filters. In general, Algorithm 3.1_HOG performs better than the state-of-the-art methods considered do for most of the challenging attributes in both the quantitative and qualitative evaluations.

Despite a good performance provided by Algorithm 3.1, there is a need to improve the tracking performance in some of the challenging situations of OTB-50, and VOT2016. In view of this, a second tracking algorithm, Algorithm 4.1, based on a structural local 2DDCT sparse appearance model and an occlusion handling mechanism has been proposed. The energy compaction property of 2DDCT has been exploited in the object representation by using only a few 2DDCT coefficients, which has reduced the computational cost of the l_1-minimization used in the algorithm. By considering only a few 2DDCT coefficients in each local patch of the dictionary and candidate samples, this algorithm requires 2.18 fps as compared with 1.86 fps required by ASLA, thereby increasing the speed of this second algorithm by 17.2% compared to that of ASLA. In contrast to the existing spatial domain object representation schemes, Algorithm 4.1 represents the object in the transform domain. A holistic image reconstruction procedure from the overlapped local patches that are obtained

using the local patch dictionary and the sparse codes has been presented. A method of generating a robust occlusion map from the reconstructed holistic image and the pooled feature vector has been obtained. A novel observation model update mechanism has been developed to handle the appearance change of the target and to avoid model degradation. A patch occlusion ratio has been introduced for the confidence score computation to enhance the tracking performance. Experiments conducted on the two datasets mentioned above bear out that the proposed Algorithm 4.1 generally performs better than the state-of-the-art tracking methods and the first tracking algorithm proposed in the thesis do for most of the challenging attributes in both the quantitative and qualitative evaluations.

In spite of the good performance of Algorithm 4.1, there are still some challenging situations in the OTB-50 and VOT2016 datasets for which further improvement in the performance of Algorithm 3.1 and Algorithm 4.1 is desired. Therefore, these issues have been addressed by proposing yet another tracking algorithm, Algorithm 5.1, based on a collaboration of the discriminative and generative appearance models. In the discriminative model, two kernelized correlation filters have been used to estimate the target location and a new generative model has been used to find the other affine motion parameters of the target. The motivation for developing this method of estimating the location and other affine motion parameters of the target in the different models has been drawn from the intuitions that (1) the discriminative capability of a tracker plays a significant role in finding the location of the target rather than in estimating the other affine motion parameters of the target and (2) the generative capability of a tracker plays an important role in finding the remaining affine motion parameters of the target. A new generative appearance model has been presented by extending the robust coding technique used in the first tracking algorithm to two dimensions, and then introducing it into the bilateral 2DPCA reconstruction procedure to develop an IRRC technique. The introduction of 2D robust coding takes care of non-Gaussian or non-Laplacian noise and avoids the effect of the outliers or occlusion

when computing the projection coefficients of the bilateral 2DPCA projection matrices. A 2D robust coding distance measure has been introduced to find the candidate sample having appearance similar to that of the reconstructed one from the bilateral 2DPCA subspace, and has been used in the observation likelihood calculation to find the tracking result. The weights obtained during IRRC have been exploited to detect outliers or occlusion as well as to generate an occlusion map that has been used in the observation model update of both the bilateral 2DPCA subspace and the two kernelized correlation filters. Algorithm 5.1 has been tested on the gray scale features (Algorithm 5.1_Gray) and HOG features (Algorithm 5.1_HOG) by conducting extensive experiments on the two datasets for performance comparison with several state-of-the-art methods and with the first two tracking algorithms (Algorithm 3.1 and Algorithm 4.1) proposed in the thesis. The proposed Algorithm 5.1_HOG generally outperforms the methods considered for most of the challenging attributes in both the quantitative and qualitative evaluations.

6.2 Scope for Future Investigation

The various schemes resulting from the research work of the thesis can be extended in different ways both to visual tracking and in other research areas of computer vision. Considering the fact that the reconstruction of the object from the linear PCA subspace may not be accurate as the candidate samples may lie beyond the range of the linear subspace during a large appearance change of the object, non-linear or kernel PCA subspace models can be investigated to model the object appearance for visual tracking. Similarly, to handle a large appearance variation of the object, the non-linear or kernel bilateral 2DPCA subspace models can be explored for visual tracking. Further, the 2DRC in both the bilateral 2DPCA and kernel bilateral 2DPCA subspace models can be investigated for face recognition. The occlusion detection mechanism using weights obtained from robust coding or 2D robust coding can also

be explored in face recognition. The PCA subspace model and its update can be performed in the 2DDCT domain instead of in the spatial domain, as in Chapter 4, to reduce the computational complexity further as the object appearance model is already in the 2DDCT domain. The reconstruction of holistic image from the overlapped local patches, proposed in Chapter 4, could be explored for applications in the areas of computer vision, where there is a need for such a reconstruction. Also, the deep features from convolutional neural networks and their combination with gray, HOG and color attribute features can be investigated to model the object appearance in the proposed tracking algorithms. A study on the inclusion of a module to re-detect the object in the proposed tracking algorithms can be carried out to address the challenges of long-term tracking or long-term full occlusion. Finally, the proposed tracking algorithms can be extended to multi-object tracking.

References

[1] D. A. Ross, J. Lim, R.-S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, vol. 77, pp. 125–141, May 2008.

[2] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Sep 2009, pp. 1436–1443.

[3] D. Wang, H. Lu, and M. H. Yang, "Online object tracking with sparse prototypes," *IEEE Trans. on Image Processing*, vol. 22, no. 1, pp. 314–325, Jan 2013.

[4] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, Dec 2006.

[5] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov 2011.

[6] S. Dubuisson and C. Gonzales, "A survey of datasets for visual tracking," *Mach. Vision and Appl.*, vol. 27, no. 1, pp. 23–52, Jan 2016.

[7] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2013, pp. 2411–2418.

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of Int. Joint Conf. on Artificial Intell. (IJCAI)*, Aug 1981, pp. 674–679.

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 25, no. 5, pp. 564–577, May 2003.

[10] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 26, no. 6, pp. 810–815, Jun 2004.

[11] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2006, pp. 798–805.

[12] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2010, pp. 1269–1276.

[13] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2012, pp. 1830–1837.

[14] W. Ou, D. Yuan, Q. Liu, and Y. Cao, "Object tracking based on online representative sample selection via non-negative least square," *Multimedia Tools and Appl.*, vol. 77, no. 9, pp. 10 569–10 587, 2018.

[15] X. Jia, H. Lu, and M. H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2012, pp. 1822–1829.

[16] J. Yang, R. Xu, J. Cui, and Z. Ding, "Robust visual tracking using adaptive local appearance model for smart transportation," *Multimedia Tools and Appl.*, vol. 75, no. 24, pp. 17 487–17 500, Dec 2016.

[17] Y. Yi, Y. Cheng, and C. Xu, "Visual tracking based on hierarchical framework and sparse representation," *Multimedia Tools and Appl.*, pp. 1–23, 2017.

[18] H. Wang, H. Ge, and S. Zhang, "Object tracking via 2DPCA and l_2-regularization," *J. of Electrical and Comput. Engineering*, vol. 2016, pp. 1–7, Jul 2016.

[19] D. Wang and H. Lu, "Object tracking via 2DPCA and l_1-regularization," *IEE Signal Processing Letters*, vol. 19, no. 11, pp. 711–714, Nov 2012.

[20] P. Qu, "Visual tracking with fragments-based PCA sparse representation," *Int. J. of Signal Processing, Image Processing and Pattern Recogn.*, vol. 7, no. 2, pp. 23–34, Feb 2014.

[21] M. Sun, D. Du, H. Lu, and L. Zhang, "Visual tracking with a structured local model," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, Sep 2015, pp. 2855–2859.

[22] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Apr 1996, pp. 329–342.

[23] M. Black, D. Fleet, and Y. Yacoob, "A framework for modeling appearance change in image sequences," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Jan 1998, pp. 660–667.

[24] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 25, no. 10, pp. 1296–1311, Oct 2003.

[25] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 29, no. 9, pp. 1661–1667, Sep 2007.

[26] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2006, pp. 260–267.

[27] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2009, pp. 983–990.

[28] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Oct 2008, pp. 234–247.

[29] S. Avidan, "Support vector tracking," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 26, no. 8, pp. 1064–1072, Aug 2004.

[30] F. Wang, J. Zhang, Q. Guo, P. Liu, and D. Tu, "Robust visual tracking via discriminative structural sparse feature," in *Proc. of the Chinese Conf. on Image and Graphics Technologies*, Jun 2015, pp. 438–446.

[31] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 27, no. 10, pp. 1631–1643, Oct 2005.

[32] S. Avidan, "Ensemble tracking," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 29, no. 2, pp. 261–271, Feb 2007.

[33] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2010, pp. 49–56.

[34] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Nov 2011, pp. 1323–1330.

[35] K. Zhang, L. Zhang, and M. H. Yang, "Real-time object tracking via online discriminative feature selection," *IEEE Trans. on Image Processing*, vol. 22, no. 12, pp. 4664–4677, Dec 2013.

[36] X. Li, A. Dick, C. Shen, A. Hengel, and H. Wang, "Incremental learning of 3D-DCT compact representations for robust visual tracking," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 35, no. 4, pp. 863–881, Apr 2013.

[37] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, vol. 14, Dec 2001, pp. 841–848.

[38] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2006, pp. 87–94.

[39] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Oct 2007, pp. 1–8.

[40] Q. Yu, T. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proc. of European. Conf. on Comput. Vision (ECCV)*, Oct 2008, pp. 678–691.

[41] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2010, pp. 723–730.

[42] W. Zhong, H. Lu, and M. H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2012, pp. 1838–1845.

[43] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. on Image Processing*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[44] C. Xie, J. Tan, P. Chen, J. Zhang, and L. He, "Collaborative object tracking model with local sparse representation," *J. of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 423–434, Feb 2014.

[45] H. Zhang, F. Tao, and G. Yang, "Robust visual tracking based on structured sparse representation model," *Multimedia Tools and Appl.*, vol. 74, no. 3, pp. 1021–1043, 2015.

[46] B. Zhuang, L. Wang, and H. Lu, "Visual tracking via shallow and deep collaborative model," *Neurocomputing*, vol. 218, pp. 61–71, Dec 2016.

[47] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image processing*, vol. 15, no. 12, pp. 3736–3745, Dec 2006.

[48] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2009, pp. 1794–1801.

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

[50] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 33, no. 11, pp. 2259–2272, Nov 2011.

[51] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Sep 2010, pp. 624–637.

[52] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient L1 tracker with occlusion detection," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2011, pp. 1257–1264.

[53] J. Yan and M. Tong, "Weighted sparse coding residual minimization for visual tracking," in *Proc. of Visual Communications and Image Processing (VCIP)*, Nov 2011, pp. 1–4.

[54] M. Jiang, H. Wang, and B. Wang, "Robust visual tracking based on maximum likelihood estimation," *Int. J. of Digital Content Tech. and its Appl.*, vol. 6, no. 22, pp. 467–474, Dec 2012.

[55] Q. Wang, F. Chen, W. Xu, and M. H. Yang, "Online discriminative object tracking with local sparse representation," in *Proc. of the IEEE Workshop on the Appl. of Comput. Vision (WACV)*, Jan 2012, pp. 425–432.

[56] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. of European Conf. on Comput. Vision (ECCV)*, May 2006, pp. 589–600.

[57] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2006, pp. 728–735.

[58] Y. Wu, J. Cheng, J. Wang, and H. Lu, "Real-time visual tracking via incremental covariance tensor learning," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Sep 2009, pp. 1631–1638.

[59] M. Chen, S. K. Pang, T.-J. Cham, and A. Goh, "Visual tracking with generative template model based on riemannian manifold of covariances," in *Proc. of the IEEE Int. Conf. on Information Fusion (FUSION)*, Jul 2011, pp. 1–8.

[60] Y. Wu, B. Wu, J. Liu, and H. Lu, "Probabilistic tracking on riemannian manifolds," in *Proc. of the IEEE Int. Conf. on Pattern Recogn. (ICPR)*, Dec 2008, pp. 1–4.

[61] Y. Wu, J. Wang, and H. Lu, "Robust bayesian tracking on riemannian manifolds via fragments-based representation," in *Proc. of the IEEE Int. Conf. Acoustics, Speech and Signal Processing, (ICASSP)*, Apr 2009, pp. 765–768.

[62] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng, "Visual tracking via incremental log-euclidean riemannian subspace learning," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2008, pp. 1–8.

[63] T. Wang, I. Gu, and P. Shi, "Object tracking using incremental 2D-PCA learning and ML estimation," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2007, pp. 933–936.

[64] M.-X. Jiang, M. Li, and H.-Y. Wang, "Visual object tracking based on 2DPCA and ML," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–7, 2013.

[65] D. Wang, H. Lu, and M.-H. Yang, "Robust visual tracking via least soft-threshold squares," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 26, no. 9, pp. 1709–1721, Sep 2016.

[66] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2010, pp. 2544–2550.

[67] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2014, pp. 1090–1097.

[68] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. of British Machine Vision Conf. (BMVC)*, Sep 2014, pp. 1–11.

[69] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 39, no. 8, pp. 1561–1575, Aug 2017.

[70] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Oct 2012, pp. 702–715.

[71] ——, "High-speed tracking with kernelized correlation filters," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 37, no. 3, pp. 583–596, Mar 2015.

[72] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2015, pp. 5388–5396.

[73] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Oct 2016, pp. 419–433.

[74] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2016, pp. 1401–1409.

[75] T. Zhang, S. Liu, C. Xu, B. Liu, and M.-H. Yang, "Correlation particle filter for visual tracking," *IEEE Trans. on Image Processing*, vol. 27, no. 6, pp. 2676–2687, Jun 2018.

[76] H. Kiani Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2015, pp. 4630–4638.

[77] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Dec 2015, pp. 4310–4318.

[78] ——, "Convolutional features for correlation filter based visual tracking," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV) Workshops*, Dec 2015, pp. 58–66.

[79] ——, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2016, pp. 1430–1438.

[80] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Sep 2014, pp. 254–265.

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

[82] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Trans. on Image Processing*, vol. 18, no. 7, pp. 1512–1523, Jul 2009.

[83] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Dec 2015, pp. 3074–3082.

[84] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV)*, Dec 2015, pp. 3119–3127.

[85] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2016, pp. 4303–4311.

[86] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Oct 2016, pp. 472–488.

[87] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2009, pp. 248–255.

[88] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. of Comput. Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015.

[89] C. He, Y. F. Zheng, and S. C. Ahalt, "Object tracking using the gabor wavelet transform and the golden section algorithm," *IEEE Trans. on Multimedia*, vol. 4, no. 4, pp. 528–538, Dec 2002.

[90] H. Sun, Q. Bu, and H. Zhang, "PSO based gabor wavelet feature extraction and tracking method," in *Proc. of SPIE*, Nov 2008, pp. 1–8.

[91] K. Selvakumar and J. Jerome, "Robust object tracking via class aware partial least squares gabor wavelet subspace," *Procedia Engineering*, vol. 64, no. 0, pp. 159 – 168, 2013.

[92] P. Prez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. of European. Conf. on Comput. Vision (ECCV)*, May 2002, vol. 2350, pp. 661–675.

[93] D. Wang, H. Lu, and Y. W. Chen, "Incremental MPCA for color object tracking," in *Proc. of the IEEE Int. Conf. on Pattern Recogn. (ICPR)*, Aug 2010, pp. 1751–1754.

[94] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, and R. Pflugfelder, "The visual object tracking VOT2016 challenge results," in *Proc. of European Conf. on Comput. Vision (ECCV)*, Oct 2016, pp. 1–45.

[95] I. Jolliffe, *Principal Component Analysis*. Springer-Verlag New York, Inc., 2002.

[96] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 22, no. 1, pp. 4–37, Jan 2000.

[97] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 12, no. 1, pp. 103–108, Jan 1990.

[98] N. F. Güler and S. Koçer, "Classification of EMG signals using PCA and FFT," *J. of Medical Systems*, vol. 29, no. 3, pp. 241–250, Jun 2005.

[99] H. Kong, L. Wang, E. K. Teoh, X. Li, J.-G. Wang, and R. Venkateswarlu, "Generalized 2D principal component analysis for face image representation and recognition," *Neural Networks*, vol. 18, no. 5, pp. 585–594, Jul 2005.

[100] 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. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 26, no. 1, pp. 131–137, Jan 2004.

[101] D. Zhang and Z.-H. Zhou, "$(2D)^2$PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 13, pp. 224 – 231, Dec 2005.

[102] M. Isard and A. Blake, "Condensation: Conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, Aug 1998.

[103] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 38, no. 11, pp. 2137–2155, Nov 2016.

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

[105] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 33, no. 8, pp. 1619–1632, Aug 2011.

[106] L. Čehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Trans. on Image Processing*, vol. 25, no. 3, pp. 1261–1274, Mar 2016.

[107] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang, "Incremental tensor subspace learning and its applications to foreground segmentation and tracking," *Int. J. Comput. Vision*, vol. 91, no. 3, pp. 303–327, Feb 2011.

[108] D. Wang, H. Lu, and C. Bo, "Visual tracking via weighted local cosine similarity," *IEEE Trans. on Cybernetics*, vol. 45, no. 9, pp. 1838–1850, Sep 2015.

[109] B. K. Shreyamsha Kumar, M. N. S. Swamy, and M. Omair Ahmad, "Robust coding in a global subspace model and its collaboration with a local model for visual tracking," *Special issue on Artificial Intelligence in Multimedia Computing, Multimedia Tools and Appl.*, pp. 1–27, 2019.

[110] ——, "Weighted residual minimization in PCA subspace for visual tracking," in *Proc. of the IEEE Int. Symp. on Circuits and Systems (ISCAS)*, May 2016, pp. 986–989.

[111] T. Zhou, H. Bhaskar, K. Xie, J. Yang, X. He, and P. Shi, "Online learning of multi-feature weights for robust object tracking," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, Sep 2015, pp. 725–729.

[112] T. Zhou, H. Bhaskar, F. Liu, J. Yang, and P. Cai, "Online learning and joint optimization of combined spatial-temporal models for robust visual tracking," *Neurocomputing*, vol. 226, pp. 221–237, Feb 2017.

[113] X. Zhang, G.-S. Xia, Q. Lu, W. Shen, and L. Zhang, "Visual object tracking by correlation filters and online learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 77–89, Jun 2018.

[114] T. Zhou, F. Liu, H. Bhaskar, and J. Yang, "Robust visual tracking via online discriminative and low-rank dictionary learning," *IEEE Trans. on Cybernetics*, vol. 48, no. 9, pp. 2643–2655, Sep 2018.

[115] M. Yang, D. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2011, pp. 625–632.

[116] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. of Comput. Vision*, vol. 26, no. 1, pp. 63–84, Jan 1998.

[117] D. Wang, H. Lu, Z. Xiao, and M.-H. Yang, "Inverse sparse tracker with a locally weighted distance metric," *IEEE Trans. on Image Processing*, vol. 24, no. 9, pp. 2646–2657, Sep 2015.

[118] B. K. Shreyamsha Kumar, M. N. S. Swamy, and M. Omair Ahmad, "Visual tracking via bilateral 2DPCA and robust coding," in *Proc. of the IEEE Canadian Conf. on Electrical and Comput. Engineering (CCECE)*, May 2016, pp. 1–4.

[119] D. Wang, H. Lu, and C. Bo, "Fast and robust object tracking via probability continuous outlier model," *IEEE Trans. on Image Processing*, vol. 24, no. 12, pp. 5166–5176, Dec 2015.

[120] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, Dec 2013, pp. 809–817.

[121] http://www.cs.toronto.edu/~dross/ivt. Accessed: Jun, 2014.

[122] http://www.dabi.temple.edu/~hbling/code/L1-APG_release.zip. Accessed: Aug, 2015.

[123] http://faculty.ucmerced.edu/mhyang/project/tip13_prototype/TIP12-SP.htm. Accessed: Mar, 2014.

[124] https://github.com/huchuanlu/15_9. Accessed: Feb, 2016.

[125] https://github.com/huchuanlu/15_12. Accessed: Feb, 2016.

[126] http://faculty.ucmerced.edu/mhyang/project/cvpr13_lss/LSST-MatlabCode-V1.zip. Accessed: Sep, 2015.

[127] https://github.com/huchuanlu/14_1. Accessed: Oct, 2015.

[128] http://winsty.net/dlt.html. Accessed: Apr, 2017.

[129] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling l1 tracker with occlusion detection," *IEEE Trans. on Image Processing*, vol. 22, no. 7, pp. 2661–2675, Jul 2013.

[130] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recogn. (CVPR)*, Jun 2011, pp. 1313–1320.

[131] P. Dai, Y. Luo, W. Liu, C. Li, and Y. Xie, "Robust visual tracking via part-based sparsity model," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 1803–1806.

[132] X. You, X. Li, Z. He, and X. Zhang, "A robust local sparse tracker with global consistency constraint," *Signal Processing*, vol. 111, pp. 308–318, Jun 2015.

[133] B. K. Shreyamsha Kumar, M. N. S. Swamy, and M. Omair Ahmad, "Visual tracking using structural local DCT sparse appearance model with occlusion detection," *Multimedia Tools and Appl.*, vol. 78, no. 6, pp. 7243–7266, 2019.

[134] ——, "Structural local DCT sparse appearance model for visual tracking," in *Proc. of the IEEE Int. Symp. on Circuits and Systems (ISCAS)*, May 2015, pp. 1194–1197.

[135] C. Lin and C.-M. Pun, "Tracking object using particle filter and DCT features," in *Proc. of Int. Conf. on Advances in Comput. Science and Engineering*, Jun 2013, pp. 167–169.

[136] H. Chen, W. Zhang, X. Zhao, and M. Tan, "DCT representations based appearance model for visual tracking," in *Proc. of the IEEE Int. Conf. on Robotics and Biometrics (ROBIO)*, Dec 2014, pp. 1614–1619.

[137] G. Feng and J. Jiang, "JPEG compressed image retrieval via statistical features," *Pattern Recogn.*, vol. 36, no. 4, pp. 977–985, Apr 2003.

[138] D. He, Z. Gu, and N. Cercone, "Efficient image retrieval in DCT domain by hypothesis testing," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, Nov 2009, pp. 225–228.

[139] B. K. Shreyamsha Kumar, M. N. S. Swamy, and M. Omair Ahmad, "Multiresolution DCT decomposition for multifocus image fusion," in *Proc. of the IEEE Canadian Conf. on Electrical and Comput. Engineering (CCECE)*, May 2013, pp. 1–4.

[140] B. K. Shreyamsha Kumar, "Multifocus and multispectral image fusion based on pixel significance using discrete cosine harmonic wavelet transform," *Signal, Image and Video Processing*, vol. 7, no. 6, pp. 1125–1143, Nov 2013.

[141] M. Shivamurti and S. Narasimhan, "Analytic discrete cosine harmonic wavelet transform (ADCHWT) and its application to signal/image denoising," in *Proc. of the IEEE Int. Conf. on Signal Processing and Communications (SPCOM)*, Jul 2010, pp. 1–5.

[142] B. K. Shreyamsha Kumar, "Image denoising using discrete cosine harmonic wavelets," Sensor Signal Process. Group, Central Research Laboratory, Bangalore, India, Tech. Rep., Jul 2010.

[143] Z. M. Hafed and M. D. Levine, "Face recognition using the discrete cosine transform," *Int. J. Comput. Vision*, vol. 43, no. 3, pp. 167–188, Jul 2001.

[144] M. Uzair, A. Mahmood, and A. S. Mian, "Hyperspectral face recognition using 3D-DCT and partial least squares." in *Proc. of British Machine Vision Conf. (BMVC)*, Sept 2013, pp. 1–10.

[145] D. Chen, Q. Liu, M. Sun, and J. Yang, "Mining appearance models directly from compressed video," *IEEE Trans. on Multimedia*, vol. 10, no. 2, pp. 268–276, Feb 2008.

[146] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 22, no. 4, pp. 385–392, Apr 2000.

[147] W. Pennerbaker and J. Mithchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[148] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar 2010.

[149] http://faculty.ucmerced.edu/mhyang/project/cvpr12_jia_project.htm. Accessed: Oct, 2013.

[150] https://github.com/chhshen/DCT-Tracking. Accessed: Dec, 2013.

[151] S. Cao, X. Wang, and K. Xiang, "Visual object tracking based on motion-adaptive particle filter under complex dynamics," *EURASIP J. on Image and Video Processing*, vol. 2017, no. 1, p. 76, 2017.

[152] B. K. Shreyamsha Kumar, M. N. S. Swamy, and M. Omair Ahmad, "Visual tracking based on correlation filter and robust coding in bilateral 2DPCA subspace," *IEEE Access*, vol. 6, pp. 73 052–73 067, 2018.

[153] S. A. Siena, "Improving the design and use of correlation filters in visual tracking," Ph.D. dissertation, Carnegie Mellon University, Dept. Elect. Comput. Eng, USA, 2017.

[154] B. Bai, Y. Li, J. Fan, C. Price, and Q. Shen, "Object tracking based on incremental Bi-2DPCA learning with sparse structure," *Applied optics*, vol. 54, no. 10, pp. 2897–2907, Apr 2015.

[155] Y. Sui, Y. Tang, L. Zhang, and G. Wang, "Visual tracking via subspace learning: A discriminative approach," *Int. J. of Comput. Vision*, vol. 126, no. 5, pp. 515–536, 2018.

[156] Y. Sui, G. Wang, L. Zhang, and M.-H. Yang, "Exploiting spatial-temporal locality of tracking via structured dictionary learning," *IEEE Trans. on Image Processing*, vol. 27, no. 3, pp. 1282–1296, Mar 2018.

[157] Y. Sui and L. Zhang, "Visual tracking via locally structured Gaussian process regression," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1331–1335, Feb 2015.

[158] Y. Sui, Y. Tang, and L. Zhang, "Discriminative low-rank tracking," in *Proc. of the IEEE Int. Conf. on Comput. Vision*, Dec 2015, pp. 3002–3010.

[159] https://scholar.harvard.edu/files/suiyao/files/ddl.zip. Accessed: Apr, 2018.

[160] https://scholar.harvard.edu/files/suiyao/files/dlr.zip. Accessed: Apr, 2018.

[161] https://scholar.harvard.edu/files/suiyao/files/lsgpr.zip. Accessed: Apr, 2018.

[162] https://scholar.harvard.edu/files/suiyao/files/dsl.zip. Accessed: Apr, 2018.

[163] http://www.cvl.isy.liu.se/research/objrec/visualtracking/colvistrack/ColorTracking_code.zip. Accessed: Jun, 2015.

[164] http://www.robots.ox.ac.uk/~joao/circulant/tracker_release2.zip. Accessed: Jun, 2015.