MOTION-AUGMENTED INFERENCE AND JOINT KERNELS IN STRUCTURED LEARNING FOR OBJECT TRACKING AND INTEGRATION WITH OBJECT SEGMENTATION

KUMARA RATNAYAKE

A DOCTORAL THESIS
IN
THE DEPARTMENT
OF
ELECTRICAL AND COMPUTER ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

AUGUST 2016

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the doctoral thesis prepared

By:	**Mr. Kumara Ratnayake**
Entitled:	**Motion-Augmented Inference and Joint Kernels in Structured Learning for Object Tracking and Integration with Object Segmentation**

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

Doctor of Philosophy (Electrical and Computer Engineering)

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

Signed by the final examining committee:

______________________	Chair	Dr. Tiberiu Popa
______________________	External Examiner	Dr. Robert Laganiere
______________________	External Examiner	Dr. Thomas Fevens
______________________	Internal Examiner	Dr. Asim J. Al-Khalili
______________________	Internal Examiner	Dr. Yousef R. Shayan
______________________	Supervisor	Dr. Maria A. Amer

Approved ____

Chair of Department or Graduate Program Director

______________ 20 ______ ______________________________________

Dean of Faculty of Engineering and Computer Science

Abstract

Motion-Augmented Inference and Joint Kernels in Structured Learning for Object Tracking and Integration with Object Segmentation

Kumara Ratnayake
Concordia University, 2016

Video object tracking is a fundamental task of continuously following an object of interest in a video sequence. It has attracted considerable attention in both academia and industry due to its diverse applications, such as in automated video surveillance, augmented and virtual reality, medical, automated vehicle navigation and tracking, and smart devices. Challenges in video object tracking arise from occlusion, deformation, background clutter, illumination variation, fast object motion, scale variation, low resolution, rotation, out-of-view, and motion blur. Object tracking remains, therefore, as an active research field. This thesis explores improving object tracking by employing *1)* advanced techniques in machine learning theory to account for intrinsic changes in the object appearance under those challenging conditions, and *2)* object segmentation.

More specifically, we propose a fast and competitive method for object tracking by modeling target dynamics as a random stochastic process, and using structured support vector machines. *First*, we predict target dynamics by harmonic means and particle filter in which we exploit kernel machines to derive a new entropy based observation likelihood distribution. *Second*, we employ online structured support vector machines to model object appearance, where we analyze responses of several kernel functions for various feature descriptors and study how such kernels can be optimally combined to formulate a single joint kernel function. During learning, we develop a probability formulation to determine model updates and use sequential minimal optimization-step to solve the structured optimization problem. We gain efficiency improvements in the proposed object tracking by *1)* exploiting particle filter for sampling the search space instead of commonly adopted dense sampling strategies, and *2)* introducing a motion-augmented regularization term during inference to constrain the output search space.

We then extend our baseline tracker to detect tracking failures or inaccuracies and re-initialize itself when needed. To that end, we integrate object segmentation into tracking. *First*, we use binary support vector machines to develop a technique to detect tracking failures (or inaccuracies) by monitoring internal variables of our baseline tracker. We leverage learned examples from our baseline tracker to train the employed binary support vector machines. *Second*, we propose an automated method to re-initialize the tracker to recover from tracking failures by integrating an active contour based object segmentation and using particle filter to sample bounding boxes for segmentation.

Through extensive experiments on standard video datasets, we subjectively and objectively demonstrate that both our baseline and extended methods strongly compete against state-of-the-art object tracking methods on challenging video conditions.

Acknowledgments

First of all, I must thank my PhD advisor Dr. Maria A. Amer for her precious mentorship over many years. Looking back, her valuable guidance has transformed me to an independent researcher. It is much harder to describe my gratitudes towards Dr. Amer as my advisor. Her taste and choice of problems, discipline and commitment to hard work, all had a great influence on me. I must deeply appreciate Dr. Amer's patience during my ups and downs, her encouragements to try new and bolder approaches and push the limits, all of which made my PhD journey much smoother and at the same time much more exciting.

I would like to thank my external examiner Dr. Robert Laganiere and my committee members Dr. Thomas Fevens, Dr. Asim J. Al-Khalili, and Dr. Yousef R. Shayan for their reading of this thesis to help improve its presentation greatly. To my fellow VidPro colleagues, I need to thank Meisam Rakhshanfar and Tarek Ghoniemy for helping me in many ways.

I would also like to thank my colleagues at Teledyne Dalsa for their support and understanding throughout this journey.

I am deeply grateful to my parents Punchibanda Ratnayake and Nandawathie Menike Ratnayake for their support during all of my life's endeavors. I must also thank my three children Melina, Gihara, and Nelith for their patience during the course of writing this *book*. Above all, I am endlessly grateful to my beloved wife Niromi Perera for her patience, support, love, and encouragement which gave me the determination required to complete this research. I must thank her for the countless scarifies she had endured for me throughout this long journey.

This thesis is dedicated to my wife Niromi.

Contents

List of Figures

List of Tables

Nomenclature

C^{CMU}	SVM model update threshold.
C^{HM}	Precision parameter of KHM GRBF.
C^{SITR}	Number of segmentation iterations.
C^{SVM}	Structured SVM regularization parameter.
H	Entropy of a distribution.
$N^{\bullet}$	Number of most recent frames used in segmentation particle filter.
N^{EX}	Number of input-output examples.
N^{HM}	Number of states in the KHM motion model.
N^{PP}	Number of prior states considered for estimating number of particles.
N^{P}	Number of particles set at the first frame in the motion model.
N^{SP}	Number of particles in segmentation particle filter.
$\alpha^{\bullet}$	Lagrangian multiplier of binary Support-Vector-Machines.
α	Lagrangian multiplier of Structured Support-Vector-Machines.
$\bar{C}^{\text{SVM}}$	Binary SVM regularization parameter.
$\bar{N}^{\text{EX}}$	Number of examples pairs in the binary Support-Vector-Machines.
$\boldsymbol{\Phi}(\mathbf{x}, \mathbf{y})$	Joint feature map of Structured Support-Vector-Machines.
$\boldsymbol{\Theta}$	Segmentation histogram feature vector.
γ^{MI}	Precision parameter of motion GRBF.
$\gamma^{\diamond}$	Precision parameter of color GRBF.
$\gamma^{\bullet}$	Precision parameter of tracking failure-detection GRBF.
$\bar{\mathbf{n}}_t$	Non-Gaussian noise process.
$\bar{\mathbf{w}}$	Weight vector of binary Support-Vector-Machines.
$\bar{\boldsymbol{\Phi}}$	Feature map of binary Support-Vector-Machines.
$\mathbf{s}$	State vector of the motion model.
$\mathbf{z}$	Observation model of particle filter.
$\mathbf{z}^{\bullet}$	Observation model of the segmentation particle filter.
$\mathbf{u}$	State vector of particle filter.
$\mathbf{s}^{\bullet}$	Origins of the bounding boxes drawn by the segmentation particle filter.
$\mathbf{k}^{\text{OL}}$	Kernel similarity vector.
$\mathbf{n}_t$	Gaussian noise model.
$\mathbf{w}$	Weight vector of Structured Support-Vector-Machines.
$\mathbf{x}^{\star}$	Input-color features.
$\mathbf{x}^{\diamond}$	Input-global features.
$\mathbf{x}^{\text{OPT}}$	Feature vector of the optimum inferred bounding box.

$\mathbf{x}^{\bullet}$	Segmentation feature descriptor.
$\mathbf{y}^{\text{OPT}}$	The optimum inferred bounding box.
$\mathcal{D}^{+}$	Set of positive support vectors.
$\mathcal{D}^{-}$	Set of negative support vectors.
$\mathcal{H}$	Reproducing Kernel Hilbert Space.
$\mathcal{M}$	Segmentation foreground mask.
$\mathcal{S}$	Set of input-output examples.
$\mathcal{X}$	Input feature space.
$\mathcal{Y}^{\star}$	Restricted output space.
$\mathcal{Y}$	Structured output space.
$\mu^{\mathbf{u}}$	Mean of the motion vectors of the previous N^{PP} particles.
ω	Weight of segmentation particle filter.
ϕ	Discriminant function.
$\sigma^{\mathbf{u}}$	Standard deviation of the motion vectors of the previous N^{PP} particles.
φ	Motion-augmented regularization term.
$\mathbf{K}$	Joint kernel matrix.
$\xi^{\bullet}$	Slack variables of binary Support-Vector-Machines.
ξ	Slack variables of Structured Support-Vector-Machines.
$b^{\bullet}$	Bias term of binary Support-Vector-Machines.
f	Prediction function.
g	Joint kernel parameter.
$k^{\bullet}$	Kernel function of binary Support-Vector-Machines.
k	Joint kernel function.
m	Particle-weight.
$p(\mathbf{k}_r^{\text{OL}} \mid \hat{\mathbf{u}}_{t-1})$	Normalized distribution of similarity vector.
$y^{\bullet}$	Class labels of binary Support-Vector-Machines.

Acronyms

BB	Bounding Box.
CAMSGPF	Continuously Adaptive Mean Shift Guided Particle Filter.
CMU	Conditional Model Update.
FPS	Frames Per Seconds.
GRBF	Gaussian Radial Basis Function.
HoG	Histogram of Oriented Gradients.
JKF	Joint Kernel Function.
KHM	Kernelized Harmonic Means.
MAP	Maximum A Posteriori.
MIST	Motion Inferred Structured Tracker.
MIST-SEG	MIST Integrated with Segmentation.
MKL	Multiple Kernel Learning.
ORB	Oriented FAST and Rotated BRIEF.
RKHS	Reproducing Kernel Hilbert Space.
SIR	Sequential Importance Resampling.
SMO	Sequential Minimal Optimization.
Structured SVMs	Structured Support-Vector-Machines.
SVMs	Support-Vector-Machines.

Chapter 1

Introduction

With rapid advances in semiconductor technologies, low cost smart devices with increasing computational power have become ubiquitous in our daily lives. Video processing is an active research area, which aims at devising advanced algorithms that enable such devices to perceive the visual world. Despite much effort on high level tasks, including object detection, recognition, and tracking, state-of-the-art video processing systems are by far inferior when compared to the human ability of understanding and interpretation of such tasks. In this thesis, we address the task of following moving object throughout a video sequence; a task often referred to as *object tracking*. The main objective of our proposed research is to build and validate a fast, effective, and automated object tracking system. More specifically, this thesis focuses on generic model-free online object tracking where no prior knowledge about the target is available, other than the target's initial selection by means of a rectangular bounding box.

1.1 Motivation

Video object tracking starts when a moving object first appears in a video scene, and typically estimates the tracked-object's position, motion, and shape. Object tracking has gained increased attention in both academia and industry as it is the core in widespread application-domains [1–4], such as

1. **augmented and virtual reality applications**, for example, entertainment, education,

medical, and manufacturing,

2. **traffic applications**, such as automatic vehicle detection to optimize traffic flowing for the existing transportation infrastructures, detecting traffic violations (speed, lights, and lane crossing etc.), identifying hazardous vehicles, and managing toll booths,
3. **security applications**, such as detecting potential hazardous problems and emergency situations pertinent to the public safety, and protecting critical infrastructures and assets, traffic violations (speed, lights, and lane crossing etc.), identifying hazardous vehicles, and managing toll booths,
4. **counting applications**, such as determining the number of clients at entrances in retail stores enabling efficient management of wait time, queue length, and service points.

Conventional tracking systems heavily depend on human operators to continuously monitor video scenes for any abnormal events and alert relevant authorities [5]. However, such human-operated systems are error prone because prolonged monitoring of colossal number of videos is tedious, exhaustive, and uninteresting. Video data are often stored and used as passive records for subsequent forensic investigations. Failing to detect critical incidents can be fatal, particularly, in security applications. Therefore, the modality of tracking systems is shifting from solely human-operated model to partially or fully automated model [1, 3].

Some system integrators provide solutions with some degree of automated tracking capabilities [3]. Such systems require strict operating conditions in carefully controlled environments. Intrinsic visual changes, for example, due to weather conditions, daylight changes, and occlusion affect tremendously the effectiveness of those systems generating frequent false alarms. Consequently, there are growing concerns on the feasibility and viability of adopting them in real-world practical applications. Moreover, these systems are built on high-performance workstations, where video data from multiple cameras are streamed in, processed, and displaced. Such server based architectures are large, heavy, power hungry, and unreliable due to slow responsiveness and long latency in communication channels limiting the overall accuracy and scalability of such systems.

1.2 Challenges

Object tracking is challenging due to camera noise in the scene, occlusion, illumination variation, fast object motion, deformation, background clutter, low resolution, scale variation, in-plane rotation, out-plane rotation, out-of-view, motion blur, and speed requirements [1,3,6–10].

Over the last few decades, many tracking methods have been proposed to overcome these challenges. Most of those hypotheses are ad-hoc to specific applications. Modeling the object's motion dynamics is important for accurate and efficient object tracking when the motion is large or abrupt. Motion dynamics is often modeled with linear systems with additive Gaussian noise. Kalman filtering technique is typically used to compute the complete statistic of such linear-Gaussian model [11,12]. However, tracking of real-world objects induces multimodal distributions which are non-linear, non-Gaussian problems [1,3,5,13]. Computing the distribution of non-linear, non-Gaussian problems analytically is intractable; thus many algorithms have been proposed to approximate them. Particle filter [14–16], which recursively estimates the posterior distribution of the state space using Monte Carlo integration, is a very popular approach to approximate non-linear and non-Gaussian problems.

Traditional trackers [17–24] without explicit appearance modeling are suboptimal under challenging conditions, such as deformation, scale variation, in-plane rotation, out-plane rotation, and illumination variation. Online appearance modeling based on machine learning is effective in taking intrinsic appearance changes into account. Online models built on machine learning theories require self-learning from past data. This is a difficult problem, specially in model-free object tracking, where no prior knowledge about the target is given other than the initial selection of the object. Many machine learning based trackers rely on a single cue–kernel combination for learning and inference. The choice of a particular cue–kernel combination depends on the context of the application. However, a single cue–kernel combination may not be reliable in all conditions. In many situations, it can degrade the tracker's accuracy. For example, color cues are effective to partial occlusion and camera noise, but can be weak when color features of the background are similar to the tracked object. Color cues are also sensitive to lighting changes. On the other hand, edge-based cues are invariant to illumination changes, but are sensitive to occlusion and camera noise. Other pitfalls of state-of-the-art machine learning based trackers include *1*) lack of effective mechanisms for target motion modeling, *2*) high computational demand required during sampling, *3*) continuous model updates even during object occlusion and background clutter, and *4*) lack of regularization to constrain the output search space. Our objective is to extend the strength of machine learning based object tracking methods by *1*) effective target dynamic modeling, *2*) formulating joint kernel functions for effective object tracking, *3*) constraining output search space during inference, and *4*) exploiting kernel machines to evaluate posterior likelihood.

Tracking failure is inevitable. The tracker may not be able to locate the target due to its own drifts in addition to intrinsic object disappears due to occlusion and out-of-view. Recovering from tracking failures is a challenging problem which requires effective mechanisms for both detecting tracking failures and re-initialization. Integrating segmentation can alleviate this problem, but segmentation is often erroneous in the presence of background clutter and occlusion. Therefore, relying on segmentation output frequently (i.e., each frame) to reduce tracking drift is undesirable. Our second objective is to improve object tracking by *1*) effectively detecting tracking failures or inaccuracies, and *2*) automatically re-initializing object tracking effectively using segmentation and particle filters in the event of tracking failures or inaccuracies.

1.3 Requirements of Effective Object Tracking Technique

Following lists some of the main requirement of an effective object tracking technique.

Automated tracking system: In order to apply object tracking to real-world applications, we require an automated tracking system. Given only the initial selection of the object of interest, the system autonomously performs tracking. This also implies that no off-line training is required, which is often the case for popular model-based trackers, such as those based on deep learning [25]. Such system requires no manual intervention in case of tracking failures and subsequent re-initialization stage.

Efficiency (speed): We require an efficient object tracking system; an aspect less focused by many state-of-the-art tracking system. Modern video cameras contain high resolution image sensors with capabilities of streaming video at high frame rates. In order for object tracking to be useful for interactive applications, the system must perform fast.

Error resiliency and reliability: Tracking drift and failure are often encountered due to inherent challenges associated with visual object tracking. Tracking systems requiring following objects for long-term must be resilient to such failures. To that end, the system must be built with effective appearance model that can optimally discriminate the object of interest from its background.

Scalability: We require a scalable object tracking platform with functionally independent components. Instead of a complex system with hardwired one-piece module, such modular system provides operator with the flexibility of trading-off tracking accuracy and efficiency.

1.4 Summary of Contributions

We started our research by investigating object detection methods and their FPGA implementations. We then moved to study object tracking methods, such as particle filter [14,26–28] and Continuously Adaptive Mean Shift Guided Particle Filter (CAMSGPF) [24]. Acknowledging the limitations of these methods, we investigated machine learning based object tracking. Finally, we integrated object segmentation into object tracking to improve accuracy.

The contributions of this thesis are:

1. A fast and competitive method for tracking video objects by modeling target dynamics and using structured support vector machines, where
   (a) we represent the target dynamics as a random stochastic process and use harmonic means and particle filter for predicting it,
   (b) we formulate a new observation likelihood model for the particle filter by using kernel machines and entropy to evaluate certainty of the likelihood distribution,
   (c) we derive an adaptive weighted joint kernel function,
   (d) we construct a probability formulation to determine selective model updates in the structured maximization problem,

   (e) we introduce a motion-augmented regularization term during inference to constrain the output search space.

2. A technique for improving the effectiveness of object tracking, where
   (a) we detect tracking failures based on online binary support vector machines framework and
   (b) we introduce an automated method to re-initialize the tracker based on an active contour based object segmentation and utilized particle filters to sample bounding boxes for segmentation.
3. A hardware implementation of object detection, where
   (a) we integrate Mixture-of-Gaussian background modeling, noise estimation, and motion detection and
   (b) we propose a new Gaussian parameter compression technique.

So far, we have published two papers [29,30], and two journal paper are being prepared (based on object tracking and integration of segmentation) to submit to IEEE Transactions on Circuits and Systems for Video Technology.

1.5 Thesis Outline

This thesis is organized as follows.

In Chapter 2, we *first* review related work on traditional object tracking methods. We *then* focus on our discussion on machine learning based object tracking techniques by categorizing them into *1*) generative, *2*) discriminative, and *3*) hybrid methods.

In Chapter 3, we *first* present our baseline object tracking method, *then*, we describe our technique of improving the effectiveness of the proposed baseline tracker by introducing a failure detection technique and integrating it with object segmentation.

In Chapter 4, we present the objective and subjective experimental results of both our baseline tracker and its integration with an active contour based object segmentation, which we have validated using large datasets by classifying the video sequences into several challenging categories.

In Chapter 5, we conclude the contributions of the thesis and discuss possible avenues for future research.

Chapter 2

Related Work

2.1 Overview

In this Chapter, we *first* review traditional object tracking methods. *Second*, we focus on our discussion on machine learning based object tracking techniques by categorizing them into *1*) generative, *2*) discriminative, and *3*) hybrid methods. *Third*, we outline recent work on integrating object segmentation into object tracking. *Finally*, we summarize the Chapter.

2.2 Traditional Object Tracking Methods

There is extensive bibliography on video object tracking. Recent advances and future trends in tracking methods are comprehensively described in surveys [1,3,6–10]. In general, moving object tracking methods can be broadly classified into three categories; interest point based, silhouette based and kernel based [1].

In interest point based tracking [22,31–33], moving objects are detected and represented by a set of interest points (for example, corners) at each frames. Tracking is performed by linking the correspondences of these points between frames. Most of the interest point based tracking methods assume that the interest points of a given object have homogeneous motion vectors, thus may fail in tracking isolated objects with interest points moving in different directions.

In silhouette based tracking [18–20,23], the complete objects are detected in every frame using the information encoded within the object region. The objects are typically modeled with contours, edges or histograms. Tracking is achieved by matching one or more of these models for the silhouettes in each frame. Silhouette trackers are generally sensitive to camera noise.

In kernel based tracking [1, 34, 35], objects are modeled using shape and appearance. For example, an object can be modeled using a rectangle as a geometric shape, and a color histogram as an appearance model. Tracking is achieved by computing the motion of each objects frame by frame. Among the three tracking categories, kernel based tracking is widely adopted due to its accuracy and computational efficiency [34]. Current research on kernel based methods primarily focuses on incorporating Kalman filter, mean-shift, and particle filter for object tracking. We review some of this research next.

Rowe et al. [36] used Kalman filter for tracking multiple objects by incorporating a block based color histogram matching method. This method involves tuning many parameters to get good performance, thus it may fail in tracking objects in complex environments. In [11,12,37], Kalman filter is adopted for object tracking in noisy environments. However, these methods lack quantitative analysis for object occlusion, so they can suffer from tracking drift problems. Mean-shift method utilizes center-weighted histograms for object tracking [34]. The weights are defined by a spatial circular kernel which gives higher weights to the pixels in the vicinity of the object center. Mean-shift method maximizes the appearance similarity iteratively by comparing the weighted histograms of the object being tracked and window around the hypothesized object location. Bhattacharyya coefficient [38, 39] is often used for histogram comparison. In [34, 40], Comaniciu et al. introduced mean-shift procedure for object tracking. Here, non-rigid moving objects are tracked under partial occlusion by maximizing the Bhattacharyya coefficient. Collins [41] extended this early work with Lindebergs theory [42], so the objects with scale change can be tracked. This method is, however, computationally expensive. In [43], Zivkovic and Krose also extended the mean-shift procedure to adapt for objects scale and shape changes. Ning et al. introduced a mean-shift tracker using the joint color-texture features in [44] . Despite the efforts, most of Kalman and mean-shift tracking drift away in the presence of rapid motion, appearance changes, complete object occlusion, and lighting variations.

Particle filter, also known as condensation [14] or Sequential Importance Sampling (SIS) [15], has been proven to be a powerful and reliable tool for moving object tracking due to its excellent effectiveness, simplicity and flexibility in adapting nonlinear and non-Gaussian systems [1,3,13]. In [14], Isard and Blake introduced condensation for object tracking, which was then extended to color based tracking [16,45]. Such methods can suffer from tracking drift problem in environments where moving object appears in similar colors to the background. In [46], Lu et al. used grids of Histogram of Oriented Gradients (HoG) as object representation to alleviate such tracking drifts.

The effectiveness of object tracking based on particle filter is theoretically improved when more particle samples utilized. However, the computational cost in the particle filter increases as the number of particle samples increase. There has been some research focusing on to reduce computational complexity in the particle filter based tracking. Zhou et al. dynamically adapted the number of particle samples based on the video noise in [26]. In this method, the number of utilized particles is directly proportional to the noise variance. In [45], Khan et al. adopted Rao-Blackwellization [47] method to analytically compute a portion of the *posterior distribution* over the state space. This has substantially decreased the number of samples required to track a moving object. Linzhou et al. [48] introduced the concept of active particles in an attempt to reduce the computational complexity of the tracker. Recently, an adaptive selection of the number of particles depending on the output of an active contour has been introduced in [49]. Although these methods require fewer particles than conventional particle filter approaches and can handle rapid object motion, they are not usually effective against cluttered background and occlusion.

A common problem with particle filter is the degeneracy phenomenon [50], that causes the variance of the particle-weights to increase over time [51]. This means that the majority of particles would have negligible weights after a few iterations, resulting in a highly skewed *posterior distribution*. Consequently, subsequent samples drawn from this skewed *posterior distribution* can deteriorate filters performance. To overcome degeneracy problem, Shan et al. [21] proposed Mean Shift Embedded Particle Filter (MSEPF) that embeds mean shift in the particle filter. MSEPF performs mean shift search on each particles and then merge particles to nearby modes with larger probability. Recently, Wang et al. [24] combined MSEPF [21] with Continuously Adaptive Mean Shift (CAMShift) [17] and proposed Continuously Adaptive Mean Shift Guided Particle Filter (CAMSGPF) to further improve the accuracy of object tracking using significantly fewer particle samples. Here, CAMShift and particle filter are exploited to optimize the position and scale of each particle. Particularly, CamShift is applied on the whole particle set in a simplified way by removing the redundancy between the particles. Moreover, CAMSGPF employs an ad-hoc scheme to more efficiently overcome particle degeneracy problem. Thus, CAMSGPF is efficient in tracking moving objects with varying scales in cluttered background, and it outperforms trackers based on conventional particle filter and MSEPF [24]. The observation model of CAMSGPF is a color histogram, which is sensitive to lighting changes and can be weak when color features of the background are similar to the tracked object. Therefore, CAMSGPF is suboptimal under severe occlusion. Integration of multiple cues has recently been applied to improve the effectiveness in some tracking systems in [27, 28], where color and motion cues are used to tackle some challenges in object tracking, such as illumination variation and background clutter.

While some effort [26, 46]) has been put forth, adaptive template updates are largely overlooked in many traditional object tracking methods reviewed above. In order to account for intrinsic appearance changes, online appearance modeling based on machine learning theories attracted a lot of attention recently [1, 7]. Our object tracking technique is an online machine learning method. In the next Section, we, therefore, focus on related work on machine learning based object tracking.

2.3 Machine Learning based Object Tracking Methods

In general, based on various appearance modeling, online machine learning based object tracking can be categorized into three classes: generative, discriminative, and hybrid generative-discriminative methods. Our method is online discriminative as it exploits Structured Support-Vector-Machines (Structured SVMs) [52] to effectively discriminate the surrounding background from the target.

2.3.1 Generative Methods

In generative object tracking methods, the object appearance is learned online to adapt to appearance changes, and object tracking is expressed as finding the most similar object to this learned appearance model [1]. The most trivial approach to model the target appearance is with a rectangle patch at the start of object tracking. Object tracking can be expressed as registering this rectangle patch in subsequent frames, by maximizing some similarity function, e.g., Bhattacharyya distance [39] of histograms between target and candidate. A major drawback of these trivial methods is that they are computationally expensive. Ross et al. developed a generative object tracking method [53] that incrementally learns a low-dimensional subspace representation to account for appearance changes. The generative methods [54–58] that are based on sparse representation have been successful in recent years. These methods represent target as a sparse linear combination of dictionary templates. These target models are updated online in order to adapt to appearance changes. In [54,55] Mei et al. used a holistic representation of the object as the appearance model and then solved the l_1 minimization problem. Another related work to solving l_1 minimization problem for object tracking was carried out by Bao et al. [56]. Sparse and discriminative set of features are used to improve the object tracking quality in [58], while histograms of the local sparse representation are incorporated with mean-shift to locate the target object in [57]. In [59], Wang et al. propose a generative object tracking scheme by maintaining holistic appearance information and representing the target in compact form. This method exploits classic principal component analysis methods [60] and sparse representation schemes [61] to learn appearance models online, therefore [59] can handle heavy occlusion in higher resolution images more efficiently. Tian et al. propose a generative tracking method using a local sparse model and particle filter to localize candidate samples in [62]. The method utilizes a hash coding scheme as a similarity to evaluate the resemblance of appearance model with target candidates. Because [62] uses least absolute shrinkage and selection operator [63] to solve sparse coefficients, the method [62] demands a high computational load. Despite the demonstrated successes of these generative object tracking methods, they are computationally expensive. Moreover, the accuracy of these methods is sub-optimal due to the lack of discriminative information in the appearance model to successfully separate the object from the background.

2.3.2 Discriminative (Tracking-by-Detection) Methods

Discriminative object tracking methods aim at computing a decision boundary that can best describe to separate the object from the background rather than explicitly modeling the object appearance as in generative methods. The discriminative object tracking methods are also referred as tracking-by-detection [64, 65], where the target and background are described by set of features at the initialization stage, and a binary classifier is used to distinguish the target from background. The classifier is updated to account for appearance changes in successive frames [65, 66]. In [67, 68], Grabner et al. presented discriminative object tracking schemes based on online boosting algorithm, that passes a labeled sample to boost through weak classifiers. However, such trackers are sensitive to noise because classifier is updated with its own classification results. Moreover, these methods predict the unlabeled samples at the initialization frame which can degrade the effectiveness in object tracking. To overcome such ambiguities, Babenko et al. [64, 69] presented an online Multiple Instance Learning (*MIL*) scheme to improve the flexibility of the classifier. The *MIL* tracker learns a discriminative classifier from positive and negative bags of samples. A positive bag of samples is generated by collecting the target Bounding Box (BB) and rectangular patches that are in very close proximity to the target. Multiple negative bags of samples are collected from rectangular patches that are far away from the target. The object location is determined by taking the highest classification score. The old classifier parameters and the new data points are used to update of the classifier. *MIL* tracker adopts dense sampling strategy to locate the target at the expense of high computational load. In [70], Zhang et al. extended the *MIL* tracker that incorporate the sample importance into the online learning scheme to recognize the samples in the positive bag. However, [64, 69, 70] update the classifiers using the positive labels for all samples in the positive bag, and such update procedure can diminish the discriminability of the classifier. In [66], Hare et al. presented Structured output object tracking framework (*Struck*) that integrates the learning and tracking without incorporating ad-hoc update strategies. *Struck* employs Support-Vector-Machines (SVMs) in its structured output framework due to their good generalization ability, effectiveness to label noise, and flexibility in object representation through the use of kernels. The appearance is modeled by Haar features [71] and intensity histograms. The target location is computed by obtaining the highest discriminant score of the classifier. To gain efficiency improvements in *Struck*, the authors incorporate a budgeting mechanism which constrains the number of support vectors. *Struck* updates the classifier with the new data derived from the current target location. Some of the limitations of *Struck* are lack of dynamic motion modeling, occlusion handling, tracking articulated and deformable objects that undergo scale variations. In [65], Kalal et al. presented discriminative classifier learning method decomposed into tracking, learning and detection (*TLD*). The method incorporates object detector with an optical flow tracker for appearance modeling, which is subsequently used for correcting any tracking drifts. The optical flow tracker is based on Lucas Kanade tracker [72, 73] which estimates the displacements of interesting points. The appearance model is based on binary patterns and Random Ferns [74] are utilized to learn the object detector. Similar to the other discriminative methods, the positive samples are drawn from the target location and negative samples are selected from locations further from the optimal target location. *TLD* evaluates the errors of object detector by using positive and negative experts which estimate missed detections and false alarms, respectively. Because *TLD* is based on interesting points, its performance is particularly suboptimal for articulated objects and when the object undergoes rotation.

Recently, correlation filters [75] have been applied to object tracking methods [76–79] to improve efficiency. In [76], Ma et al. propose a tracker based on discriminative correlation filters by decomposing the task of object tracking into translation and scale estimation problems. To reduce tracking drifts, the method [76] uses an online random fern based classifier for re-detecting any objects. In [77], Henriques et al. exploit circulant structures in natural images and uses Fourier transforms to reduce storage and computational demand in their kernelized correlation filter (*KCF*) tracker. The method [77] assumes that it can train a classifier efficiently from background patches, but this can produce unwanted boundary effects thereby degrading the accuracy of *KCF*. Danelljan et al. in [78] relax this assumption and extends *KCF* by introducing a regularization term to penalize coefficients of correlation filter taking corresponding spatial location into account. Efficiency improvements are achieved in [78] by using low-dimensional color features. Because the correlation filter based trackers [77, 78] limit the utilization of a single kernel, Tang et al. propose a method in [79] that incorporate multiple kernels. Nevertheless, despite superior performance in [76–79], correlation filter based trackers often drifts away from the target during affine transformations or non-rigid deformations.

A graph based discriminative tracker is proposed in [80], which uses tensors to model appearance of the target. In [80], geometric structure of object and its background are differentiated by dedicating multiple graphs. The method reduces the tensor dimensions by exploiting graph embedding. The technique [80] is semi-supervised therefore restricting its use in limited applications. Liang et al. adopt BING objectness [81] for object tracking in [82], where BING is adapted individually for each videos and objects being tracked. The method [82], however, requires off-line training using SVMs. In [83], a discriminative tracker based on cognitive dynamic systems is proposed. The method [83] utilizes feedback and feed-forward mechanisms to effectively track small objects and uses Kalman filters to infer the target locations. Experimental results reported in [83] are limited to few video sequences and the method is compared with few state-of-the-art trackers.

2.3.3 Hybrid Generative-Discriminative Methods

When sufficient training data is available, the discriminative methods often outperforms the generative methods. However, if the training data is scarce, generative models often have better generalization performance [84]. Hybrid generative and discriminative methods are combined to benefit from both types of methods [85–87]. Zhong et al. [87] has developed sparsity-based discriminative classifier and a sparsity-based generative model. Here, the discriminative model computes a confidence value that assigns more weights to the foreground than the background, while the histogram-based generative model incorporates spatial information to handle occlusion. In [86], Wand et al. has presented a hybrid scheme based on an over-complete dictionary to represent local image patches of target, and then learned a classifier to separate the object from the background. Here, targets are searched by using the over-complete dictionary in a high-dimensional feature space, requiring high computational cost. Another hybrid method based on discriminative naïve Bayes classifier and a static random projection matrix [88] is proposed in [85]. Here, the appearance model is extracted from compressed domain features, and the discriminative classifier is updated with positive and negative samples drawn from the current frame. However, such methods based on static random projection matrix can introduce tracking drifts in dynamic video sequences with large appearance changes. An online dictionary based discriminative tracker in which target appearance is modeled by sparse representation is proposed in [89]. The method constrains a sparsity consistency term to exploit properties of generative and discriminative of the appearance. In [89], partial occlusion is handled by constraining an elastic net to capture local appearance characteristics. In general, the hybrid models require tuning many parameters to trade-off the overall influence between generative and discriminative models, thus improper hybrid models can be of worse performance than native generative or discriminative models [90].

2.4 Integration of Object Segmentation into Tracking

Majority of recent work on integration of segmentation into tracking can be grouped based on the employed segmentation methods. In what follows, we summarize some of the recent work on object segmentation integrated into object tracking based on graph-cuts, active contours, Random walk, and watershed.

Object tracking integrated with graph cut segmentation are presented in [91–94]. In [91], the authors present a marker-less object tracking method for augmented reality applications. By integrating graph cut segmentation within the optical flow tracker, the result of [91] show that the integrated method is effective in tracking articulated objects under challenging environmental conditions. In [92], Malcolm et al. propose multiple object tracking framework by fusing graph cut technique as segmentation method. The method spatially constrains the object segmentation process to a user defined object region so that accurate segmentation can be performed. Papadakis et al. in [93], decompose object into visible and occluded regions which are tracked assuming the velocity of each object can be represented by a dynamic model. Graph cut segmentation is employed to separate predicted regions which allows handling partial and full occlusions. A discriminative tracker with a rough segmentation based on graph cut is proposed in [94], which aims to track deformable target with a discriminative classifier. The method requires a ground truth bounding box of the object at the start of each video to initialize the tracker. However, graph cut based segmentation methods are prone to errors in the presence of background clutter and occlusion.

Object segmentation based on active contours [95] are fused with tracking in [96–99]. Paragios et al. propose a multiple object tracking framework minimizing a geodesic active contour objective model using stochastic gradient descent in [96]. Using a Gaussian mixture model, motion detection is performed. The method tracks complex contours and efficiently handles topological changes for the evolving contours with a computationally efficient implementation. In [97], Zhou et al. present an object tracker in which shape priors are incorporated and modeled using active contours. The method requires off-line training stage where a shape codebook representing the shape mode is trained. In [98], object foreground is segmented from the background using an active contour scheme that preserves accurate object boundaries. The method exploits the extracted boundaries and learns dynamic shape models that enable effective tracking during occlusion. Another object tracking method by active contour segmentation is presented in [99], which uses a level set to represent the object and utilizes the Bhattacharyya distance [38] to locate the region that optimally describes object being tracked. The active contour segmentation is used to refine the contours of the target. In [96–99], segmentation output is integrated with tracking in each frame. During background clutter or occlusion, segmentation result is often unreliable; consequently, integrating the segmentation with tracking at every frame can, in fact, degrade the accuracy of the tracker. Our method integrates the segmentation [100], which is also an active contour based technique, however, different from [96–99], we do not execute segmentation at every frame, but only during a tracking failure. Therefore, the speed of segmentation method does not significantly affect the overall speed of our object tracking integrated with segmentation. Also, possible segmentation errors are much less propagated into the tracking.

Random-walk [101] based segmentation is integrated in [102–104]. In [102, 103], the authors propose a color histogram based object tracking method in which Random-walk based image segmentation is utilized to track non-rigid objects. Reliable tracking is achieved by exploiting the spatial properties of the segmented object to initialize the tracking method.

Kwon et al. in [104] integrate result of semi-supervised object segmentation to enhance their object tracking method, which is based on a local patch-based appearance model. Using a deterministic local optimizer, the computational complexity of [104] is significantly reduced. These methods are suboptimal under large object displacements and complete occlusions. Moreover, [104] is semi-supervised which precludes using it for many real world applications requiring real-time performance.

Integrating segmentation to improve object tracking using Kalman filtering is proposed in [105]. The authors employ watershed [106] as the object segmentation technique and demonstrate the object tracking performance in applications such as head and hand tracking. The method is confronted with challenges, such as divergence in iterations, computational cost, over-segmentation, etc.

2.5 Summary

Object tracking is a difficult problem due to many challenges inherited in video sequences. Thus, it will continue to be an active field of research. Classical trackers without explicit appearance modeling (such as CAMShift and MSEPF) are suboptimal under challenging conditions. In order to account for intrinsic appearance changes, online appearance modeling based on machine learning theories has attracted significant attention. Among several categories in machine learning based object tracking, discriminative methods perform better because they compute a decision boundary that can best separate the object from the background; therefore, much of current research focus on this category.

Our object tracking method extends discriminative machine-learning based method by *1*) representing the target dynamics as a random stochastic process and use harmonic means and particle filter for predicting it, *2*) formulating a new observation likelihood model for the particle filter by using kernel machines and entropy to evaluate certainty of the likelihood distribution, *3*) developing an adaptive weighted joint kernel function to construct an effective appearance model, *4*) devising a probability formulation to determine model updates and for structured maximization problem, and *5*) formulating a motion-augmented regularization term during inference to constrain the output search space.

To improve object tracking, intuitively, object segmentation can be exploited. Recent methods focus on explicitly integrating graph-cuts and active contours based segmentation methods into object tracking in each frame. Choosing the segmentation method and mechanism for tracking failure detection plays an important effect on overall accuracy of object tracking. The main difference between our proposed integration of segmentation into object tracking is that we do not apply each frame but only if we detect object tracking failures.

Chapter 3

The Proposed Object Tracking

3.1 Introduction

In this Chapter, we *first* outline the proposed method of modeling the target dynamics by harmonic means and particle filter, and describe our technique of using online Structured SVMs to the tracking problem, where *1*) we derive an adaptive weighted joint kernel function, *2*) we construct a probability formulation to determine model updates and for structured maximization problem, and *3*) we introduce a motion-augmented regularization term during inference to constrain the output search space. In the *second* part of the Chapter, we describe our technique of improving the effectiveness of the proposed object tracking by introducing a failure detection technique and integrating an active contour based segmentation method. Note that in Section 3.3 we propose how to use object segmentation output to improve the accuracy of object tracking. We are not studying here the interesting aspect of improving object segmentation based on object tracking such as in [107–110].

3.2 The Proposed Motion Inferred Structured Tracker

3.2.1 Algorithm Overview

Figure 1 and Algorithm 1 summarize the proposed MIST, which consists of two main steps: dynamic modeling (Section 3.2.2) and tracking by learning (Section 3.2.3). The dynamic

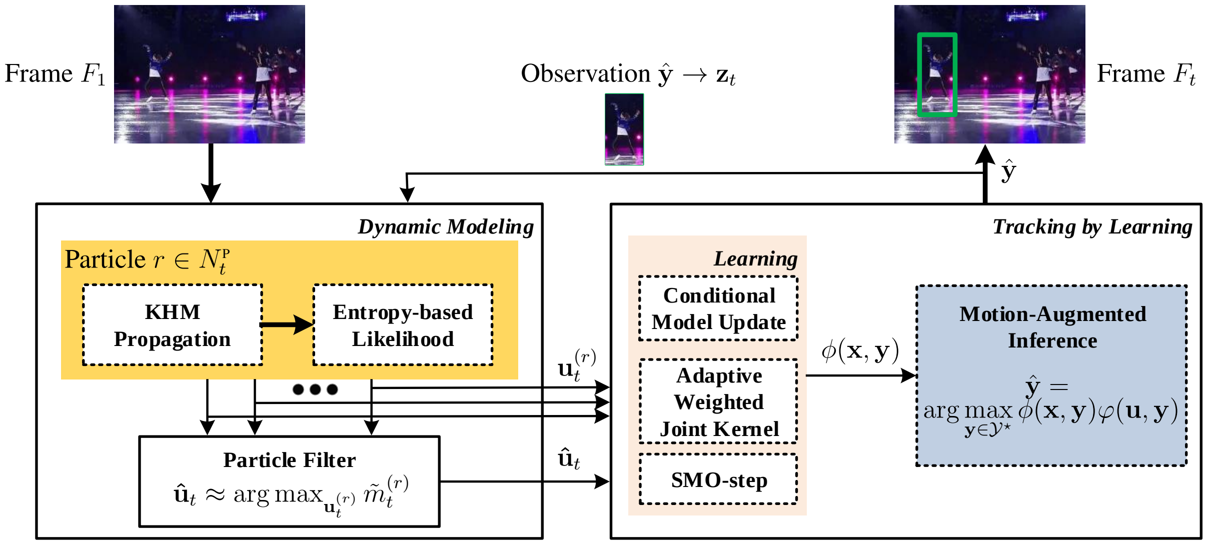
Frame F_1
Observation $\hat{\mathbf{y}} \to \mathbf{z}_t$
Frame F_t
$\hat{\mathbf{y}}$
Dynamic Modeling
Particle $r \in N_t^{\mathrm{P}}$
KHM Propagation
Entropy-based Likelihood
$\mathbf{u}_t^{(r)}$
Particle Filter
$\hat{\mathbf{u}}_t \approx \arg\max_{\mathbf{u}_t^{(r)}} \tilde{m}_t^{(r)}$
$\hat{\mathbf{u}}_t$
Tracking by Learning
Learning
Conditional Model Update
Adaptive Weighted Joint Kernel
SMO-step
$\phi(\mathbf{x}, \mathbf{y})$
Motion-Augmented Inference
$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}^\star} \phi(\mathbf{x}, \mathbf{y})\varphi(\mathbf{u}, \mathbf{y})$

Figure 1: Overview of our object tracking method. $\mathbf{u}$ is the state vector of particle filter; $\mathbf{z}$ is observation to the particle filter; $\mathbf{y}$ is output BB; $\mathbf{x}$ is feature vector of $\mathbf{y}$; $\tilde{m}$ is normalized importance weight; and $\hat{\mathbf{u}}$ and $\hat{\mathbf{y}}$ denote the optimal estimation of $\mathbf{u}$ and $\mathbf{y}$, respectively.

modeling step is composed of three modules: Kernelized Harmonic Means (KHM), particle filter, and entropy-based likelihood; the tracking by learning step consists of three modules: conditional model-update, Joint Kernel Function (JKF), and motion-inferred inference. In the dynamic modeling step, first, we use KHM to propagate the target's state dynamics and second, employ particle filter for sampling and filtering the propagated states. Third, to evaluate the certainty of the likelihood distribution of the particle filter, we formulate a new observation likelihood model by using kernel machines and entropy. Finally, applying Maximum A Posteriori (MAP) rule [51], we obtain the estimated state of the target. In tracking by learning step, first, the learning component outputs a scoring function and retain a pool of positive and negative samples. Positive and negative samples contain and describe variations of object and background, respectively. Here, we design an effective JKF using color and HoG [111] features. Then, we employ a conditional model-update scheme to minimize tracking drifts and use Sequential Minimal Optimization (SMO)-step for maximization problem. Finally, in the regularized inference step, we predict an optimal output of target BB by maximizing the scoring function is regularized with a motion-augmented term.

Algorithm 1: Proposed online object tracking algorithm

```
   Input  : Initial BB of the target in $F_1$.
   Output: Prediction of the output BB $\hat{\mathbf{y}}$ in $F_t$.
1  repeat
2      if $F_1$ then
3          Draw $N_1^{\text{P}}$ particles around the initial BB.
4          go to line 12.
5      for each particle $r$ do
6          Propagate particles according to (4).
7          Evaluate entropy-based likelihood $p(\mathbf{z}_t|\mathbf{u}_t^{(r)})$ by (16).
8          Update the importance weights $m_t^{(r)}$ using (11).
9      Estimated optimal particle state $\hat{\mathbf{u}}_t$ with (17).
10     Compute number of particles: $N_t^{\text{P}} = \mu^{\mathbf{u}} + 3\sigma^{\mathbf{u}}$.
11     Sort $\{\mathbf{u}_t^{(r)}, m_t^{(r)}\}$ according to $m_t^{(r)}$, and resample.
12     for each particle $r$ do
13         Extract color $\mathbf{x}^{\star}$ and local shape $\mathbf{x}^{\diamond}$ features.
14         Compute scoring function $\phi(\mathbf{x}, \mathbf{y})$ by (28).
15         if $\min(p(\mathcal{D}^{+}, |\mathbf{y}^{\text{OPT}}), \phi(\mathbf{x}^{\text{OPT}}, \mathbf{y})) > C^{\text{CMU}}$ then
16             Evaluate (35), and compute JKF:
17                 $k(\mathbf{x}, \bar{\mathbf{x}}) = g \cdot k^{\star}(\mathbf{x}^{\star}, \bar{\mathbf{x}}^{\star}) + (1-g) \cdot k^{\diamond}(\mathbf{x}^{\diamond}, \bar{\mathbf{x}}^{\diamond})$.
18             Compute regularization term $\varphi(\mathbf{u}, \mathbf{y})$ using (37).
19             Evaluate $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}^{\star}} \phi(\mathbf{x}, \mathbf{y})\varphi(\mathbf{u}, \mathbf{y})$.
20             Select $\{\mathbf{y}^{+}, \mathbf{y}^{-}\}$ according to (29).
21             Maximize the dual (24) using SMO-style [112].
22         else
23             Derive output BB $\hat{\mathbf{y}}$ using (4).
24 until end of video sequence.
```

3.2.2 Target Dynamic Modeling

Modeling the target dynamics is important for accurate and efficient object tracking especially when the motion is large or abrupt. The proposed dynamic modeling is composed of three modules: KHM, particle filter, and entropy-based likelihood, which are explained in the next three sections.

3.2.2.1 Kernelized Harmonic Means Propagation

We represent our dynamic model by a state vector $\mathrm{s} = [s_1 \;\; s_2]$ that describes the moving target by its $[s_1 \;\; s_2]$ position in the $2D$ Cartesian coordinate system. Typically, the dynamics of s is represented by a constant velocity model [1,7,24]

$$\hat{\mathbf{s}}_{t+1} = \mathbf{s}_t + (\mathbf{s}_t - \mathbf{s}_{t-1}) + \mathbf{n}_t^{\text{HM}}, \tag{1}$$

where $\hat{\mathbf{s}}_{t+1}$ is the predicted state at time $t+1$ (in frame F_{t+1}) and $\mathbf{s}_t$ and $\mathbf{s}_{t-1}$ are the current and previous states at time t and $t-1$, respectively, and $\mathbf{n}_t^{\text{HM}}$ is system noise modeled by a Gaussian distribution with zero mean and a standard deviation of 0.25. In (1), $(\mathbf{s}_t - \mathbf{s}_{t-1})$ represents constant velocity component computed using two most recent states. We use N^{HM} prior state vectors (e.g., $N^{\text{HM}} = 8$) and apply KHM to estimate $\Delta\mathbf{s}_t = \mathbf{s}_t - \mathbf{s}_{t-1}$ as

$$\begin{aligned}\Delta\mathbf{s}_t &= \frac{1}{\frac{k^{\text{HM}}(t-1,t-1)}{\Delta\mathbf{s}_{t-1}} + \frac{k^{\text{HM}}(t-1,t-2)}{\Delta\mathbf{s}_{t-2}} + \cdots + \frac{k^{\text{HM}}(t-1,t-N^{\text{HM}})}{\Delta\mathbf{s}_{t-N^{\text{HM}}}}} \\ &= \frac{1}{\sum\limits_{l=1}^{N^{\text{HM}}} \frac{k^{\text{HM}}(t-1,t-l)}{\Delta\mathbf{s}_{t-l}}},\end{aligned} \tag{2}$$

where $l \in [1, N^{\text{HM}}]$ is the prior state number and k^{HM} is a Gaussian Radial Basis Function (GRBF) kernel with parameter C^{HM} (e.g., $C^{\text{HM}} = 1.0$)

$$k^{\text{HM}}(t,t') = \exp\left(-C^{\text{HM}} \cdot (t-t')^2\right). \tag{3}$$

Since $k^{\text{HM}}(t,t')$ is higher for the most recent state vectors and lower for past state vectors, the later state dynamics are aggregated more (than the former dynamics) into the KHM predicted state vector. Substituting $\Delta\mathbf{s}_t$ in (1), we predict the state dynamics of the target at time $t+1$ by

$$\hat{\mathbf{s}}_{t+1} = \mathbf{s}_t + \frac{1}{\sum\limits_{l=1}^{N^{\text{HM}}} \frac{k^{\text{HM}}(t-1,t-l)}{\Delta\mathbf{s}_{t-l}}} + \mathbf{n}_t^{\text{HM}}. \tag{4}$$

We utilize the proposed state dynamics model for propagating each particle in our particle filtering process.

3.2.2.2 Particle Filter

During the initialization of the tracker (i.e., on frame F_1), we draw N_1^{P} (e.g., $N_1^{\text{P}} = 1000$) particles around the initial BB according to a Gaussian distribution. We denote the state of particle r at time t by $\mathbf{u}_t^{(r)}$ and propagate each particles with independent KHM propagation model, i.e., $\mathbf{u}_t^{(r)} \to \hat{\mathbf{s}}_t^{(r)}$, where $r \in [1, N_t^{\text{P}}]$ is particle r. The measurement to the particle filter is $\mathbf{z}_t$, which we obtain from the optimal BB estimated by the proposed SVMs inference model (see Section 3.2.3.4). We consider state dynamic prediction as an estimation problem of the system state $\mathbf{u}_t$ using a sequence of noisy measurement $\mathbf{z}_t$ made on the system. In particle filtering, state $\mathbf{u}_t$ is modeled as a Markovian random process and observation $\mathbf{z}_t$ are assumed to be conditionally independent given the state sequence. Under these assumptions, state and observation models are described [50] by

$$\left.\begin{aligned} \mathbf{u}_t &= f^{\text{ST}}(\mathbf{u}_{t-1}, \bar{\mathbf{n}}_t^{\text{S}}), \\ \mathbf{z}_t &= f^{\text{OB}}(\mathbf{u}_t, \bar{\mathbf{n}}_t^{\text{O}}). \end{aligned}\right\} \tag{5}$$

Both $f^{\text{ST}}(\cdot)$ and $f^{\text{OB}}(\cdot)$ are nonlinear functions and $\bar{\mathbf{n}}_t^{\text{S}}$ and $\bar{\mathbf{n}}_t^{\text{O}}$ are independent non-Gaussian noise processes. We are interested in making an inference about $\mathbf{u}_t$ given all the observations up to time t, $\mathbf{z}_{1:t} = \{\mathbf{z}_1, ..., \mathbf{z}_t\}$. This is given by the posterior distribution for $\mathbf{u}_t$, $p(\mathbf{u}_t|\mathbf{z}_{1:t})$ [50], which by Bayes rule

$$p(\mathbf{u}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{u}_t)p(\mathbf{u}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}. \tag{6}$$

Here, $p(\mathbf{z}_t|\mathbf{u}_t)$ is the observation likelihood distribution describing how the observation $\mathbf{z}_t$ depends on state $\mathbf{u}_t$. We assume that system dynamics are governed by a first order Markov process; thus the prior distribution $p(\mathbf{u}_t|\mathbf{z}_{1:t-1})$ can be described by the Chapman-Kolmogorov Equation [50]

$$p(\mathbf{u}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{u}_t|\mathbf{u}_{t-1})p(\mathbf{u}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{u}_{t-1}, \tag{7}$$

where $p(\mathbf{u}_t|\mathbf{u}_{t-1})$ is defined as the state transition distribution. Substituting (7) in (6), we obtain

$$p(\mathbf{u}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{u}_t)\int p(\mathbf{u}_t|\mathbf{u}_{t-1})p(\mathbf{u}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{u}_{t-1}}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}. \tag{8}$$

Equation (8) forms the optimal Bayesian solution for inferring the system state. However, (8) cannot be analytically solved [50]. Particle filter provides a numerical approximation for the posterior distribution $p(\mathbf{u}_t|\mathbf{z}_{1:t})$ using a discrete set of weighted samples (particles) $\mathbf{u}_t^{(r)}$. With such particles, the posterior density in (8) [15, 50] is approximated

$$p(\mathbf{u}_t|\mathbf{z}_{1:t}) \approx \sum_{r=1}^{N_t^{\mathrm{P}}} \tilde{m}_t^{(r)} \cdot \delta(\mathbf{u}_t - \mathbf{u}_t^{(r)}); \quad \tilde{m}_t^{(r)} = \frac{m_t^{(r)}}{\sum_{\bar{r}=1}^{N_t^{\mathrm{P}}} m_t^{(\bar{r})}}, \tag{9}$$

where $\delta(\cdot)$ is the Dirac delta function, $\tilde{m}_t^{(r)}$ is the normalized importance weight of particle r at frame F_t, and $m_t^{(r)}$ [15] is calculated

$$m_t^{(r)} \propto \frac{p(\mathbf{z}_t|\mathbf{u}_t^{(r)})p(\mathbf{u}_t^{(r)}|\mathbf{u}_{t-1}^{(r)})}{q(\mathbf{u}_t^{(r)}|\mathbf{u}_{t-1}^{(r)}, \mathbf{z}_{1:t})} m_{t-1}^{(r)}, \tag{10}$$

where $q(\mathbf{u}_t^{(r)}|\mathbf{u}_{t-1}^{(r)}, \mathbf{z}_{1:t})$ is the importance distribution from which particles are drawn at each frame. As with widely adopted Sequential Importance Resampling (SIR) particle filters [113], we choose $q(\mathbf{u}_t^{(r)}|\mathbf{u}_{t-1}^{(r)}, \mathbf{z}_{1:t})$ as the state transition distribution, i.e., $q(\mathbf{u}_t^{(r)}|\mathbf{u}_{t-1}^{(r)}, \mathbf{z}_{1:t}) = p(\mathbf{u}_t|\mathbf{u}_{t-1}^{(r)})$, and (10) becomes

$$m_t^{(r)} \propto p(\mathbf{z}_t|\mathbf{u}_t^{(r)}) \cdot m_{t-1}^{(r)}. \tag{11}$$

For improving the efficiency of the proposed tracker, we use the mean $\mu^{\mathbf{u}}$ and the standard deviation $\sigma^{\mathbf{u}}$ of the motion vectors of the previously estimated N^{PP} optimal particles (e.g., $N^{\mathrm{PP}} = 16$) to adaptively derive the number of particles N_t^{P}. To that end, we set the number of particles to the number of locations covered by a circular region with a radius of $\mu^{\mathbf{u}} + 3\sigma^{\mathbf{u}}$; more specifically, $N_t^{\mathrm{P}} = \lceil \pi(\mu^{\mathbf{u}} + 3\sigma^{\mathbf{u}})^2 \rceil$, where $\lceil \cdot \rceil$ rounds up to the nearest integer. The weights $m_t^{(r)}$ are sorted in ascending order and resampled so that the particles with higher weights are multiplied if $N_t^{\mathrm{P}} > N_{t-1}^{\mathrm{P}}$ and those with lower weights are eliminated otherwise.

3.2.2.3 Entropy-based Likelihood

The observation likelihood model $p(\mathbf{z}_t|\mathbf{u}_t)$ plays an important role in estimating the state $\mathbf{u}_t$. Entropy can be used to measure certainty of distribution; a lower entropy means less uncertainty in the underline distribution. We formulate the observation likelihood using entropy of the similarity-distribution between each particle and the target by exploiting *kernel machines* described in Section 3.2.3.3. Let $\mathbf{k}^{\text{OL}}$ be a vector of similarity between previous target $\hat{\mathbf{u}}_{t-1}$and each particle

$$\mathbf{k}^{\text{OL}} = \big[k(\mathbf{u}_1, \hat{\mathbf{u}}_{t-1}) \quad k(\mathbf{u}_2, \hat{\mathbf{u}}_{t-1}) \quad \cdots \quad k(\mathbf{u}_{N_t^{\text{P}}}, \hat{\mathbf{u}}_{t-1})\big], \tag{12}$$

where JKF $k(\mathbf{u}, \hat{\mathbf{u}}) = g \cdot k^{\star}(\mathbf{u}^{\star}, \hat{\mathbf{u}}^{\star}) + (1-g) \cdot k^{\diamond}(\mathbf{u}^{\diamond}, \hat{\mathbf{u}}^{\diamond})$, $k^{\star}$ and $k^{\diamond}$ are color and shape kernel functions, and g is a weight. To compute entropy of the similarity scores, we normalize each elements in (39)

$$\tilde{\mathbf{k}}_r^{\text{OL}} = \frac{\mathbf{k}_r^{\text{OL}} - \min(\mathbf{k}^{\text{OL}})}{\max(\mathbf{k}^{\text{OL}}) - \min(\mathbf{k}^{\text{OL}})}, \tag{13}$$

and deduce their similarity distribution $p(\mathbf{k}_r^{\text{OL}}|\hat{\mathbf{u}}_{t-1})$

$$p(\mathbf{k}_r^{\text{OL}}|\hat{\mathbf{u}}_{t-1}) = \frac{\tilde{\mathbf{k}}_r^{\text{OL}}}{\sum\limits_{r=1}^{N_t^{\text{P}}} \tilde{\mathbf{k}}_r^{\text{OL}}}. \tag{14}$$

Then, we compute the corresponding entropy score H for the distribution $p(\mathbf{k}_r^{\text{OL}}|\hat{\mathbf{u}}_{t-1})$ by

$$H = -\sum_{r=1}^{N_t^{\text{P}}} p(\mathbf{k}_r^{\text{OL}}|\hat{\mathbf{u}}_{t-1}) \log\big(p(\mathbf{k}_r^{\text{OL}}|\hat{\mathbf{u}}_{t-1})\big), \tag{15}$$

and subsequently we define our observation likelihood

$$p(\mathbf{z}_t|\mathbf{u}_t^{(r)}) = \frac{\exp(-\tilde{\mathbf{k}}_r^{\text{OL}} \cdot H)}{\sum\limits_{\acute{r}=1}^{N_t^{\text{P}}} \exp(-\tilde{\mathbf{k}}_{\acute{r}}^{\text{OL}} \cdot H)}. \tag{16}$$

By substituting (16) in the weight computation in (11), and using MAP rule [51], we obtain the estimated state $\hat{\mathbf{u}}_t$

$$\hat{\mathbf{u}}_t \approx \arg\max_{\mathbf{u}_t^{(r)}} \tilde{m}_t^{(r)}. \tag{17}$$

We incorporate state dynamics to efficiently infer the prediction within our SVMs learning model, termed as *motion-augmented inference*, as shown in Section 3.2.3.4.

3.2.3 Tracking by Learning

In general, the objective of a tracker is to maintain an estimate of the position of the target object. The tracker typically extracts and compares features from an image patch within the estimated BB and its example pairs which are usually learned online. Structured SVMs are widely used in machine learning and computer vision as they possess good generalization ability with built-in flexibility in object representation through the use of kernels, while being effective against estimation noise. Because of such rich characteristics of Structured SVMs, we utilize Structured SVMs to the object tracking problem. The proposed tracking by learning is composed of four modules: conditional model-update, joint kernels, motion-inferred inference, and SMO-step. We start by presenting classical Structured SVMs theory, in which SMO-step is outlined, and then, we describe the three remaining modules.

3.2.3.1 Structured Support-Vector-Machines

The Structured SVMs traditionally are used for classification problem, where the task is to take a set of training examples and learn a classification function to make binary labels ± 1 [114]. Instead, object tracking is considered as learning a *prediction* function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps the space of input features $\mathcal{X}$ to the space of output BBs $\mathcal{Y}$ based on N^{EX} input-output example pairs $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_{N^{\text{EX}}}, \mathbf{y}_{N^{\text{EX}}})\} \in (\mathcal{X} \times \mathcal{Y})^{N^{\text{EX}}}$. With Structured SVMs, we discriminatively learn a *scoring* function $\phi : \mathcal{X} \times \mathcal{Y} \in \mathbb{R}$ over input-output example set $\mathcal{S}$. Alternatively, the scoring function ϕ maps both output BB $\mathbf{y}$ and its corresponding feature $\mathbf{x}$ to a scalar label. Hence, ϕ can be seen as measuring the compatibility of an input-output pairs (notice that each BB $\mathbf{y}$ is extracted from the corresponding particle $\mathbf{u}$). Once the scoring function is learned, the prediction of the output $\hat{\mathbf{y}}$ that constitutes the highest compatibility with the input $\mathbf{x}$ can be obtained by maximizing ϕ over all possible output $\mathbf{y} \in \mathcal{Y}$

$$\hat{\mathbf{y}} = f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \phi(\mathbf{x}, \mathbf{y}). \tag{18}$$

The scoring function is defined [52] in the form of

$$\phi(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \boldsymbol{\Phi}(\mathbf{x}, \mathbf{y}) \rangle, \tag{19}$$

where the weight vector $\mathbf{w}$ is learned with sequentially obtained example pairs in set $\mathcal{S}$, and $\boldsymbol{\Phi}(\mathbf{x}, \mathbf{y})$ is a joint feature map that maps joint input feature and output BBs to a transform space. The specific form of $\boldsymbol{\Phi}(\mathbf{x}, \mathbf{y})$ depends on the nature of the problem. In general, $\boldsymbol{\Phi}(\mathbf{x}, \mathbf{y})$ is not explicitly modeled allowing us to exploit the advantages of kernel machines [114]. In (19), the inner product $\langle \cdot, \cdot \rangle$ is defined in a high (potentially infinite) dimensional vector space $\mathcal{H}$ referred as the Reproducing Kernel Hilbert Space (RKHS) [115], where the classes are hoped to be linearly separable.

Following the standard SVMs derivation [114], the scoring function ϕ can be learned by minimizing the constrained convex objective function

$$\left.\begin{array}{l}
\min\limits_{\mathbf{w}} \left\{ \dfrac{1}{2} \parallel \mathbf{w} \parallel^2 + C^{\text{SVM}} \sum\limits_{i=1}^{N^{\text{EX}}} \xi_i \right\}, \\
\text{subject to} \\
\forall i : \ \xi_i \geq 0 \text{ and} \\
\forall i \, \forall \mathbf{y} \neq \mathbf{y}_i : \langle \mathbf{w}, \delta \boldsymbol{\Phi}_i(\mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i,
\end{array}\right\} \tag{20}$$

where $i \in [1, N^{\text{EX}}]$, the slack variables ξ_i allow the examples to violate the constraint of being outside of the margin, $\delta \boldsymbol{\Phi}_i(\mathbf{y}) = \boldsymbol{\Phi}(\mathbf{x}_i, \mathbf{y}_i) - \boldsymbol{\Phi}(\mathbf{x}_i, \mathbf{y})$, and C^{SVM} is a parameter (e.g., $C^{\text{SVM}} = 25$) which controls how strongly margin violations are penalized. The loss function Δ is 1 when the BBs defined by $\bar{\mathbf{y}}$ and $\mathbf{y}$ are disjoint (i.e., $\bar{\mathbf{y}} \neq \mathbf{y}$), and is 0 when the BBs are identical.

Instead of solving the *primal* optimization problem in (20) directly, its *dual* formulation using the Lagrangian function [52] is obtained

$$\left.\begin{aligned}&\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{\substack{i\\ \mathbf{y}\neq\mathbf{y}_i}}\sum_{\substack{j\\ \bar{\mathbf{y}}\neq\mathbf{y}_j}}\alpha_{i\mathbf{y}}\alpha_{j\bar{\mathbf{y}}}\langle\delta\boldsymbol{\Phi}_i(\mathbf{y}),\delta\boldsymbol{\Phi}_j(\bar{\mathbf{y}})\rangle+\sum_{\substack{i\\ \mathbf{y}\neq\mathbf{y}_i}}\Delta(\mathbf{y}_i,\mathbf{y})\alpha_{i\mathbf{y}},\\ &\text{subject to}\\ &\forall i:\ \sum_{\mathbf{y}\neq\mathbf{y}_i}\alpha_{i\mathbf{y}}\leq C^{\text{SVM}}\text{ and}\\ &\forall i\,\forall\mathbf{y}\neq\mathbf{y}_i:\alpha_{i\mathbf{y}}\geq 0,\end{aligned}\right\}\tag{21}$$

where $j \in [1, N^{\text{EX}}]$ is an index, the Lagrangian multiplier $\boldsymbol{\alpha}$ corresponds to the margin constraint in (20). By solving this dual optimization problem, the weight vector $\mathbf{w} = \sum_{i,\mathbf{y}\neq\mathbf{y}_i}\alpha_{i\mathbf{y}}\delta\boldsymbol{\Phi}_i(\mathbf{y})$ and the scoring function in (19) can be rewritten

$$\phi(\mathbf{x},\mathbf{y})=\sum_{i,\bar{\mathbf{y}}\neq\mathbf{y}_i}\alpha_{i\bar{\mathbf{y}}}\langle\delta\boldsymbol{\Phi}_i(\bar{\mathbf{y}}),\boldsymbol{\Phi}(\mathbf{x},\mathbf{y})\rangle.\tag{22}$$

Following [116], we use $\boldsymbol{\beta}$

$$\beta_{i\mathbf{y}}=\begin{cases}-\alpha_{i\mathbf{y}}, & \text{if } \mathbf{y}\neq\mathbf{y}_i\\ \sum_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\alpha_{i\bar{\mathbf{y}}}, & \text{otherwise,}\end{cases}\tag{23}$$

and substitute $\boldsymbol{\beta}$ in (21) to form a simplified dual problem

$$\left.\begin{aligned}&\max_{\boldsymbol{\beta}} -\frac{1}{2}\sum_{i,\mathbf{y}}\sum_{j,\bar{\mathbf{y}}}\beta_{i\mathbf{y}}\beta_{j\bar{\mathbf{y}}}K_{ij}-\sum_{i\mathbf{y}}\Delta(\mathbf{y}_i,\mathbf{y})\beta_{i\mathbf{y}},\\ &\text{subject to}\\ &\forall i:\ \sum_{\mathbf{y}}\beta_{i\mathbf{y}}=0\text{ and}\\ &\forall i\,\forall\mathbf{y}:\beta_{i\mathbf{y}}\leq C^{\text{SVM}}\Delta'(\mathbf{y}_i,\mathbf{y}),\end{aligned}\right\}\tag{24}$$

where $\Delta'(\mathbf{y}_i, \mathbf{y}) = 1$, if $\mathbf{y} = \mathbf{y}_i$, and $\Delta'(\mathbf{y}_i, \mathbf{y}) = 0$ otherwise, and

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_1) & \cdots & k(\mathbf{x}_{N^{\text{EX}}}, \mathbf{x}_1) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_{N^{\text{EX}}}, \mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_{N^{\text{EX}}}, \mathbf{x}_1) & k(\mathbf{x}_{N^{\text{EX}}}, \mathbf{x}_2) & \cdots & k(\mathbf{x}_{N^{\text{EX}}}, \mathbf{x}_{N^{\text{EX}}}) \end{bmatrix}. \tag{25}$$

The JKF $k : \mathcal{X} \times \mathcal{X} \in \mathbb{R}$ is the inner product of input-output pairs $(\mathbf{x}_i, \mathbf{y})$ and $(\mathbf{x}_j, \bar{\mathbf{y}})$ mapped in RKHS $\mathcal{H}$ space, i.e.,

$$k\big((\mathbf{x}_i, \mathbf{y}), (\mathbf{x}_j, \bar{\mathbf{y}})\big) = \langle \mathbf{\Phi}(\mathbf{x}_i, \mathbf{y}), \mathbf{\Phi}(\mathbf{x}_j, \bar{\mathbf{y}}) \rangle. \tag{26}$$

In (24), the loss function $\Delta(\mathbf{y}_i, \mathbf{y})$ quantifies how well the estimated BB $\mathbf{y}$ approaches the output BB $\mathbf{y}_i$. Hence, it plays an important role in optimizing the maximization problem in (24). As in [117], we use the BB overlap ratio

$$\Delta(\mathbf{y}_i, \mathbf{y}) = 1 - \frac{\mathbb{A}(\mathbf{y}_i \cap \mathbf{y})}{\mathbb{A}(\mathbf{y}_i \cup \mathbf{y})}. \tag{27}$$

where $\mathbb{A}(\mathbf{y}_i \cap \mathbf{y})$ is the area of the intersection of the BBs $\mathbf{y}_i$ and $\mathbf{y}$, and $\mathbb{A}(\mathbf{y}_i \cup \mathbf{y})$ is the area of their union.

We extract the feature (color and shape) inputs $\mathbf{x}$ from their corresponding BBs $\mathbf{y}$ as in [117]. Hence, without loss of generality, we denote the JKF as $k(\mathbf{x}_i, \mathbf{x}_j)$ omitting $\mathbf{y}$. Substituting $\boldsymbol{\alpha}$ in (22) with $\boldsymbol{\beta}$ in (23), a simplified form of ϕ [116] is obtained

$$\phi(\mathbf{x}, \mathbf{y}) = \sum_{i, \bar{\mathbf{y}}} \beta_{i\bar{\mathbf{y}}} k(\mathbf{x}_i, \mathbf{x}). \tag{28}$$

Often, $\boldsymbol{\beta}$ is sparse, i.e., most of the elements in $\boldsymbol{\beta}$ have the value 0. We denote the pairs $(\mathbf{x}_i, \mathbf{y})$ for which $\beta_{i\mathbf{y}} \neq 0$ as support vectors. Support vectors with $\beta_{i\mathbf{y}} > 0$ and $\beta_{i\bar{\mathbf{y}}} < 0$ are referred as positive and negative support vectors respectively.

We adopt SMO-style [112] for maximization of our dual problem in (24) because of its proven simplicity and efficiency [116]. We select a pair of positive and negative BBs by searching for the maximum and minimum of the gradient of (24), respectively. For example, $\mathbf{y}^+$ is chosen by finding the most important positive sample according to

$$\mathbf{y}^+ = \arg\max_{\boldsymbol{y}} \ -\phi(\mathbf{x}_i, \mathbf{y}) - \Delta(\mathbf{y}_i, \mathbf{y}). \tag{29}$$

For this pair of $\mathbf{y}^+$ and $\mathbf{y}^-$, we optimize their corresponding coefficients $\beta^+_{i\mathbf{y}}$ and $\beta^-_{i\mathbf{y}}$ using SMO-step. If these coefficients are non-zero, we retain $\beta_{i\mathbf{y}}$, the corresponding gradients and support vector $(\mathbf{x}_i, \mathbf{y})$. During tracking, both gradient and $\beta_{i\mathbf{y}}$ are updated, and we remove any support vector if its $\beta_{i\mathbf{y}}$ becomes zero. In practice, however, the target may not be present in the scene, for example, due to occlusion. Therefore, updating the model every frame results in tracking drift. To alleviate this problem, we construct a probability model to determine the state of the target and effectively update our learning model, which is described next.

3.2.3.2 Conditional Model Update

During SMO-step, we search for the maximum and minimum of the gradient to select a pair of positive and negative BBs. If the corresponding coefficients ($\beta^+_{i\mathbf{y}}$ and $\beta^-_{i\mathbf{y}}$) of these positive and negative BBs are non-zero, we retain the sample as a support vector in each frame. Let $\mathbf{x}^{\text{OPT}}$ and $\mathcal{D}^+$ be the feature vector of the optimally inferred BB $\mathbf{y}^{\text{OPT}}$ and the set of positive support vectors, respectively. We define the probability that $\mathbf{y}^{\text{OPT}}$ belongs to $\mathcal{D}^+$ by

$$p(\mathcal{D}^+|\mathbf{y}^{\text{OPT}}) = \frac{\mu^+}{\mu^+ \, |\mu^-|}, \tag{30}$$

where the sum of kernel scores (μ^+ and μ^-) for positive and negative BBs are given by

$$\mu^+ = \sum_{i \in \mathcal{D}^+} \beta^+_{i\mathbf{y}} k(\mathbf{x}_i, \mathbf{x}^{\text{OPT}}), \quad \mu^- = \sum_{i \in \mathcal{D}^-} \beta^-_{i\mathbf{y}} k(\mathbf{x}_i, \mathbf{x}^{\text{OPT}}). \tag{31}$$

We retain the current estimated BB as a positive support vector only when $\min(p(\mathcal{D}^+, |\mathbf{y}^{\text{OPT}}), \phi(\mathbf{x}^{\text{OPT}}, \mathbf{y})) > C^{\text{CMU}}$ (e.g., $C^{\text{CMU}} = 0.02$). When the target is absent from the scene, $\min(p(\mathcal{D}^+, |\mathbf{y}^{\text{OPT}}), \phi(\mathbf{x}^{\text{OPT}}, \mathbf{y}))$ is lower than C^{CMU}, and we avoid updating the learning model and rely on the proposed motion model for trajectory estimation.

Moreover, to adapt Structured SVMs for object tracking, it is crucial to carefully design the JKF $k(\mathbf{x}, \mathbf{y}, \bar{\mathbf{x}}, \bar{\mathbf{y}}) = \langle \mathbf{\Phi}(\mathbf{x}, \mathbf{y}), \mathbf{\Phi}(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \rangle$ for the optimization problem in (24). Next, we discuss the proposed joint kernel design.

3.2.3.3 Adaptive Weighted Joint Kernel

For the joint kernel formulation, the input feature vector $\mathbf{x}$ is extracted from the image regions defined by the BB $\mathbf{y}$. We define $\mathbf{x} = [\mathbf{x}^{\star}\ \mathbf{x}^{\diamond}]$ by two feature descriptors to characterize the target using color $\mathbf{x}^{\star}$, and local shape $\mathbf{x}^{\diamond}$. For color features, we use joint Bhattacharyya (Hellinger) kernel [38] function

$$k^{\star}(\mathbf{x}^{\star}, \bar{\mathbf{x}}^{\star}) = \sqrt{\langle \mathbf{x}^{\star}, \bar{\mathbf{x}}^{\star} \rangle}. \tag{32}$$

By incorporating the shape kernels within the JKF, we ensure that input-output pairs with good *geometric*-similarity are assigned with higher similarity score. We construct the local shape kernel $k^{\diamond}$ as GRBF kernel function with parameter $\gamma^{\diamond} = 0.22$

$$k^{\diamond}(\mathbf{x}^{\diamond}, \bar{\mathbf{x}}^{\diamond}) = \exp\left(-\gamma^{\diamond} \parallel \mathbf{x}^{\diamond} - \bar{\mathbf{x}}^{\diamond} \parallel^{2}\right). \tag{33}$$

We design our JKF ensuring that the input-output pairs are similar *if and only if* both their inputs and outputs are similar. To this end, we define our joint kernel by taking weighted sum of color and global shape kernels. The final JKF yields a smaller output if any one of the two kernels' response is small. This also implies that the JKF is stronger than classical kernels defined over single kernel only. Formally, given two input-output pairs $(\mathbf{x}, \mathbf{y})$ and $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$, we define adaptive weighted JKF

$$\begin{aligned} k(\mathbf{x}, \bar{\mathbf{x}}) &= \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\bar{\mathbf{x}}) \rangle \\ &= g \cdot k^{\star}(\mathbf{x}^{\star}, \bar{\mathbf{x}}^{\star}) + (1-g) \cdot k^{\diamond}(\mathbf{x}^{\diamond}, \bar{\mathbf{x}}^{\diamond}) \end{aligned} \tag{34}$$

where $k^{\star}$ and $k^{\diamond}$ are color and global-shape kernel functions measuring similarity between two feature vectors. The weight $0 \leq g \leq 1$ balances the two terms and is computed adaptively by taking the color similarity between the target and its background into account:

$$g = \begin{cases} g_n; & L_n < k^\star(\mathbf{x}^\star, \bar{\mathbf{x}}^\star) \leq L_{n+1}, \end{cases} \tag{35}$$

where $\{L_n\}_{n=0}^{\infty}$ is a monotonically increasing sequence. Experimentally, we obtain $L = \{0, 0.10, 0.25, 0.50, 0.75, 1\}$ and $G = \{0.0, 0.15, 0.25, 0.35, 0.45\}$.

Note that a kernel derived by weighted sum of valid kernels holds Mercer's condition [114]. Therefore, (34) is still a valid kernel and we can evaluate the quality of complex relationship between the feature descriptors derived from both color and global shape. In particular, (34) returns with higher responses to pairs with similar features while lower responses to dissimilar features. We utilize our joint kernel in formulating the proposed observation likelihood model $p(\mathbf{z}_t|\mathbf{u}_t)$ described in Section 3.2.2. In Chapter 4, we justify the selection of features and their corresponding kernels.

However, JKF based on multiple features is computationally expensive. To gain performance improvements, we introduce a motion-augmented regularization term during inference to constrain the output search space in the next section.

3.2.3.4 Motion-Augmented Inference

In the maximization problem in (18), we infer the prediction of the output $\hat{\mathbf{y}}$ by maximizing ϕ over all possible output $\mathbf{y} \in \mathcal{Y}$, which is intractable. Instead, we leverage our proposed dynamic model and restrict the search space to a smaller subspace $\mathcal{Y}^\star \subset \mathcal{Y}$. To this end, we amend our original maximization problem in (18) with a regularization term $\varphi(\mathbf{u}, \mathbf{y})$ derived from our dynamic model

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}^\star} \ \phi(\mathbf{x}, \mathbf{y}) \cdot \varphi(\mathbf{u}, \mathbf{y}). \tag{36}$$

We want the regularization term $\varphi(\mathbf{u}, \mathbf{y})$ to be higher (smaller) if a BB $\mathbf{u}$ is closer to (far from) the output $\mathbf{y}$. To reflect this, we compute the relative distance of the dynamic models between the BBs $\mathbf{u}$ and $\mathbf{y}$. We use a distance measure based on l_2-norm of the dynamic model mapped in the RKHS space, which is subsequently induced by the JKF k^{MI}

$$\begin{aligned}\varphi(\mathbf{u},\mathbf{y}) &= \exp\big(- \parallel \mathbf{\Phi}(\mathbf{u},\mathbf{y}) - \mathbf{\Phi}(\hat{\mathbf{u}},\hat{\mathbf{y}}) \parallel^2\big) \\ &= \exp\big(2\langle\mathbf{\Phi}(\mathbf{u},\mathbf{y}),\mathbf{\Phi}(\hat{\mathbf{u}},\hat{\mathbf{y}})\rangle - \\ &\quad \langle\mathbf{\Phi}(\mathbf{u},\mathbf{y}),\mathbf{\Phi}(\mathbf{u},\mathbf{y})\rangle - \langle\mathbf{\Phi}(\hat{\mathbf{u}},\hat{\mathbf{y}}),\mathbf{\Phi}(\hat{\mathbf{u}},\hat{\mathbf{y}})\rangle\big) \\ &= \exp\big(2k^{\text{MI}}(\mathbf{u},\hat{\mathbf{u}}) - k^{\text{MI}}(\mathbf{u},\mathbf{u}) - k^{\text{MI}}(\hat{\mathbf{u}},\hat{\mathbf{u}})\big).\end{aligned} \tag{37}$$

The regularization term $\varphi(\mathbf{u},\mathbf{y})$ incorporated in (36) is not an explicit component of the learning model in (18). Therefore, it simplifies the overall learning task to a greater deal since it allows focusing on a restricted, smaller subspace $\mathcal{Y}^{\star}$. For efficiency and simplicity, we formulate our JKF for motion states k^{MI} using GRBF kernel with parameter $\gamma^{\text{MI}} = 0.2$

$$k^{\text{MI}}(\mathbf{u},\hat{\mathbf{u}}) = \exp\big(- \gamma^{\text{MI}} \parallel \mathbf{u} - \hat{\mathbf{u}} \parallel^2 \big). \tag{38}$$

3.3 The Proposed Object Segmentation Integration in MIST

In this Section, we first review the segmentation integrated with our MIST. Then, we describe the proposed integration of object segmentation into MIST (MIST-SEG), which consists of two main steps: *1*) detection of tracking failures and *2*) using object segmentation to effectively reinitialize the object tracking after a tracking failure.

3.3.1 Segmentation Method Selection

Object segmentation aims at separating perceptually relevant foreground objects from the background [100, 118–120]. We have tested the proposed MIST with three different segmentation methods: active contour-based method [100], Lazy snapping method [118], and K-means segmentation method [119]. Based on our experimental results presented in Section 4.4.2 of Chapter 4, we have selected the active contour-based segmentation method [100] due to its effectiveness and efficiency in segmenting objects. The segmentation method in [100] localizes region-based active contour energies. The energy model is composed of global, local and regularization terms. In order to retain object boundary details, the global energy term is used. The authors improve the accuracy of segmentation of images with non-homogeneous intensity regions by taking local image information (local energy term) into account. The regularization term is included to avoid the presence of small isolated segments. The method is developed using curve evolution, local statistical function, and level set techniques. Exploiting the difference of evolving contour length, the method employs a termination condition to minimize long iteration process. The experimental results presented by the authors of [100] show that the segmentation accuracy is less sensitive to its parameters and variations in the initial location of the contours (i.e., the initial BB encompassing the foreground object of interest to be segmented).

3.3.2 The Proposed Tracking Failure Detection

The integration of object segmentation requires tracking failure detection mechanism. To that end, we analyze internal variables of the proposed tracker of Section 3.2. Intuitively, one can use a temporal-statistical metric of the optimal scoring function $\phi(\mathbf{x}, \hat{\mathbf{y}})$ in (19) to determine tracking failures. However, we experimentally found that such trivial methods are ineffective. Therefore, to determine tracking failures, we propose to form a failure-detection feature vector $\mathbf{x}^\bullet$ by utilizing the following internal variables: scoring function $\phi(\mathbf{x}, \hat{\mathbf{y}})$ in (19), responses from JKF $k(\mathbf{x}, \bar{\mathbf{x}})$ in (34), color kernel $k^\star(\mathbf{x}^\star, \bar{\mathbf{x}}^\star)$ in (45), global shape kernel $k^\diamond(\mathbf{x}^\diamond, \bar{\mathbf{x}}^\diamond)$ in (32), and motion kernel $k^{\text{MI}}(\mathbf{u}, \hat{\mathbf{u}})$ in (38). Formally

$$\mathbf{x}^\bullet = \big[\phi(\mathbf{x}, \hat{\mathbf{y}}) \quad k(\mathbf{x}, \bar{\mathbf{x}}) \quad k^\star(\mathbf{x}^\star, \bar{\mathbf{x}}^\star) \quad k^\diamond(\mathbf{x}^\diamond, \bar{\mathbf{x}}^\diamond) \quad k^{\text{MI}}(\mathbf{u}, \hat{\mathbf{u}})\big]. \tag{39}$$

3.3.2.1 Failure Detection as an SVM Classification Problem

Given a set of $\bar{N}^{\text{EX}}$ example pairs $\{\mathbf{x}^\bullet_{\bar{i}}, y^\bullet_{\bar{i}}\}_{\bar{i}=1}^{\bar{N}^{\text{EX}}}$, where $\bar{i} \in [1, \bar{N}^{\text{EX}}]$ is an index, $y^{\text{S}}_{\bar{i}} \in \{-1, +1\}$ is the class label of the feature vector $\mathbf{x}^\bullet_{\bar{i}}$. We employ a standard binary SVMs model as a classifier to effectively detect the tracking failure as follows

$$\left.\begin{aligned}
&\min_{\bar{\mathbf{w}}} \left\{ \frac{1}{2} \parallel \bar{\mathbf{w}} \parallel^2 + \bar{C}^{\text{SVM}} \sum_{\bar{i}=1}^{\bar{N}^{\text{EX}}} \xi_{\bar{i}}^{\bullet} \right\}, \\
&\text{subject to} \\
&\forall \bar{i} : y_{\bar{i}}^{\bullet}(\langle \bar{\mathbf{w}}, \bar{\boldsymbol{\Phi}}(\mathbf{x}_{\bar{i}}^{\bullet}) \rangle + b^{\bullet}) \geq 1 - \xi_{\bar{i}}^{\bullet} \ \text{ and} \\
&\forall \bar{i} : \ \xi_{\bar{i}}^{\bullet} \geq 0, \quad b^{\bullet} \in \mathbb{R},
\end{aligned}\right\} \tag{40}$$

where the weight vector $\bar{\mathbf{w}}$ is learned with sequentially obtained $\bar{N}^{\text{EX}}$ example pairs $\{\mathbf{x}_{\bar{i}}^{\bullet}, y_{\bar{i}}^{\bullet}\}$, $\bar{\boldsymbol{\Phi}}(\mathbf{x}_{\bar{i}}^{\bullet})$ is a nonlinear function that maps $\mathbf{x}_{\bar{i}}^{\bullet}$ to a high-dimensional feature space, the slack variables $\xi_{\bar{i}}^{\bullet}$ allow the examples to violate the constraint of being outside of the margin, $b^{\bullet}$ is the bias term of the separating hyperplane, and $\bar{C}^{\text{SVM}}$ is a parameter (e.g., $\bar{C}^{\text{SVM}} = 25$) which controls how strongly margin violations are penalized.

Using the Lagrangian function [52], the corresponding dual expression of the optimization problem in (40) is obtained

$$\left.\begin{aligned}
&\max_{\boldsymbol{\alpha}^{\bullet}} - \frac{1}{2} \sum_{\bar{i}}^{\bar{N}^{\text{EX}}} \sum_{\bar{j}}^{\bar{N}^{\text{EX}}} \alpha_{\bar{i}}^{\bullet} \alpha_{\bar{j}}^{\bullet} y_{\bar{i}}^{\bullet} y_{\bar{j}}^{\bullet} k^{\bullet}\big(\mathbf{x}_{\bar{i}}^{\bullet}, \mathbf{x}_{\bar{j}}^{\bullet}\big) + \sum_{\bar{i}}^{\bar{N}^{\text{EX}}} \alpha_{\bar{i}}^{\bullet}, \\
&\text{subject to} \\
&\forall \bar{i} : \ \sum_{\bar{i}}^{\bar{N}^{\text{EX}}} \alpha_{\bar{i}}^{\bullet} y_{\bar{i}}^{\bullet} = 0 \text{ and} \\
&\forall \bar{i} : \bar{C}^{\text{SVM}} \geq \alpha_{\bar{i}}^{\bullet} \geq 0,
\end{aligned}\right\} \tag{41}$$

where $\bar{j} \in [1, \bar{N}^{\text{EX}}]$ is an index, the Lagrangian multiplier $\boldsymbol{\alpha}^{\bullet}$ corresponds to the margin constraint in (40), and $k^{\bullet}\big(\mathbf{x}_{\bar{i}}^{\bullet}, \mathbf{x}_{\bar{j}}^{\bullet}\big)$ is a kernel function defined as

$$k^{\bullet}\big(\mathbf{x}_{\bar{i}}^{\bullet}, \mathbf{x}_{\bar{j}}^{\bullet}\big) = \langle \bar{\boldsymbol{\Phi}}(\mathbf{x}_{\bar{i}}^{\bullet}), \bar{\boldsymbol{\Phi}}(\mathbf{x}_{\bar{j}}^{\bullet}) \rangle. \tag{42}$$

By solving this dual optimization problem in (41) using SMO-step [112], we obtain

$$\bar{\mathbf{w}} = \sum_{\bar{i}}^{\bar{N}^{\mathrm{EX}}} \alpha_{\bar{i}}^{\bullet} y_{\bar{i}}^{\bullet} \bar{\mathbf{\Phi}}(\mathbf{x}_{\bar{i}}^{\bullet}). \tag{43}$$

Once the optimization problem for SVMs has been solved, we can predict any tracking failures, i.e., $y^{\bullet}(\mathbf{x}^{\bullet}) = -1$, using the failure-detection feature vector $\mathbf{x}^{\bullet}$ corresponding to the current frame as follows

$$y^{\bullet}(\mathbf{x}^{\bullet}) = \begin{cases} +1, & \text{if} \quad \sum_{\bar{i}}^{\bar{N}^{\mathrm{EX}}} \alpha_{\bar{i}}^{\bullet} y_{\bar{i}}^{\bullet} k^{\bullet}\left(\mathbf{x}_{\bar{i}}^{\bullet}, \mathbf{x}^{\bullet}\right) + b^{\bullet} \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{44}$$

For detecting tracking failures, we construct the kernel $k^{\bullet}\left(\mathbf{x}_{\bar{i}}^{\bullet}, \mathbf{x}_{\bar{j}}^{\bullet}\right)$ as a GRBF kernel function with parameter $\gamma^{\bullet} = 1$

$$k^{\bullet}\left(\mathbf{x}_{\bar{i}}^{\bullet}, \mathbf{x}_{\bar{j}}^{\bullet}\right) = \exp\left(-\gamma^{\bullet} \parallel \mathbf{x}_{\bar{i}}^{\bullet} - \mathbf{x}_{\bar{j}}^{\bullet} \parallel^{2}\right). \tag{45}$$

For online object tracking, the proposed tracking failure detection technique must be trained online, which we describe in the next Section.

3.3.2.2 Online Training of Binary SVM for Tracking Failure Detection

We assume the target can be correctly tracked for the first $\bar{N}^{\mathrm{EX}}$ few frames (for example $\bar{N}^{\mathrm{EX}} = 32$). This is not a strong assumption since most of state-of-the-art trackers are effective in tracking the object during the first few frames. In each $\bar{N}^{\mathrm{EX}}$ frames, we leverage the proposed MIST to select a pair of positive and negative BBs by searching for the maximum and minimum of the gradient of (24), respectively. We extract the failure-detection feature vector $\mathbf{x}^{\bullet}$ corresponding to the positive and negative BBs, and use them to train the binary SVMs classifier online. Once the training is complete, i.e., after the first $\bar{N}^{\mathrm{EX}}$ frames, we predict the tracking failures using (44). In the event of any tracking failures, we re-initialize the tracker by effectively incorporating object segmentation, which we describe in the next Section.

3.3.3 The Proposed Re-initialization of MIST by Particle Filter

Often, the input to the object segmentation is provided by human labeling of an object of interest. Such manual interventions preclude the use of object segmentation to offline applications. In this work, we propose a technique based on particle filter that automatically provides the input (i.e., the initial BB) to the object segmentation. To that end, we leverage object trajectory within the most recent $N^\bullet$ frames and we use particle filter to sample and estimate the location of the optimal BB. Once tracking failure is detected, restricting the sampling to the vicinity of the most recent object trajectory is important to prevent outliers caused by background clutter. In what follows, we first discuss our method of BB selection for segmentation with particle filter (Section 3.3.3.1), and then present how we use object segmentation to effectively evaluate the observation likelihood model of particle filter (Section 3.3.3.2).

3.3.3.1 Sampling for segmentation

Let the state vector $\mathbf{s}^\bullet = [s_1^\bullet \;\; s_2^\bullet \;\; s_3^\bullet \;\; s_4^\bullet]$ describe the origin $[s_1^\bullet \;\; s_2^\bullet]$ and the width and height $[s_3^\bullet \;\; s_4^\bullet]$ of the BBs. We can regard making inference about $\mathbf{s}^\bullet$ as the estimation of the system state given a series of $\bar{t}$ observations $\mathbf{z}^\bullet{}_{1:\bar{t}} = \{\mathbf{z}^\bullet{}_1, ..., \mathbf{z}^\bullet{}_{\bar{t}}\}$. Our goal is to recursively find the posterior distribution $p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{z}^\bullet{}_{1:\bar{t}})$ for $\mathbf{s}^\bullet{}_{\bar{t}}$. Using Bayes rule [50]

$$p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{z}^\bullet{}_{1:\bar{t}}) \propto p(\mathbf{z}^\bullet{}_{\bar{t}}|\mathbf{s}^\bullet{}_{\bar{t}})p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{z}^\bullet{}_{1:\bar{t}-1}), \tag{46}$$

where the observation likelihood distribution $p(\mathbf{z}^\bullet{}_{\bar{t}}|\mathbf{s}^\bullet{}_{\bar{t}})$ describes how the observation $\mathbf{z}^\bullet{}_{\bar{t}}$ depends on $\mathbf{s}^\bullet{}_{\bar{t}}$, i.e., the origin and size of the BB. As with our motion modeling presented in Section 3.2.2.2, we assume the system dynamics can be modeled by a first order Markov process. Using Chapman-Kolmogorov Equation [50], the posterior distribution $p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{z}^\bullet{}_{1:\bar{t}})$ is calculated

$$p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{z}^\bullet{}_{1:\bar{t}}) \propto p(\mathbf{z}^\bullet{}_{\bar{t}}|\mathbf{s}^\bullet{}_{\bar{t}}) \int p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{s}^\bullet{}_{\bar{t}-1})p(\mathbf{s}^\bullet{}_{\bar{t}-1}|\mathbf{z}^\bullet{}_{1:\bar{t}-1})d\mathbf{s}^\bullet{}_{\bar{t}-1}, \tag{47}$$

where $p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{s}^\bullet{}_{\bar{t}-1})$ is the state transition distribution. We leverage the proposed KHM dynamic model in (4) to propagate the state $\mathbf{s}^\bullet$. The posterior distribution $p(\mathbf{s}^\bullet{}_{\bar{t}}|\mathbf{z}^\bullet{}_{1:\bar{t}})$ is approximated by particle filtering using a set N^{SP} weights $\omega_{\bar{t},\ell}$ corresponding to the state $\mathbf{s}^\bullet{}_{\bar{t},\ell}$, where $\ell \in$

$[1, N^{\text{SP}}]$. Using SIR particle filters [113], $\omega_{\bar{t},\ell}$ is estimated by

$$\omega_{\bar{t},\ell} \propto p(\mathbf{z}^{\bullet}{}_{\bar{t}}|\mathbf{s}^{\bullet}{}_{\bar{t},\ell}) \cdot \omega_{\bar{t}-1,\ell}. \tag{48}$$

Using MAP rule [51], the estimated location and the size of the BB $\hat{\mathbf{s}}^{\bullet}_{\hat{t}}$ is obtained from

$$\hat{\mathbf{s}}^{\bullet}_{\hat{t}} \approx \arg\max_{\mathbf{s}^{\bullet}{}_{\bar{t},\ell}} \tilde{\omega}_{\bar{t},\ell}, \tag{49}$$

where $\tilde{\omega}_{\bar{t},\ell}$ is the normalized weight. The observation likelihood model $p(\mathbf{z}^{\bullet}{}_{\bar{t}}|\mathbf{s}^{\bullet}{}_{\bar{t},\ell})$ is implicitly required to estimate $\hat{\mathbf{s}}^{\bullet}_{\hat{t}}$ in (49). In the next Section, we present our techniques of using object segmentation to effectively estimate the observation likelihood model $\omega_{\bar{t},\ell}$.

3.3.3.2 Observation likelihood model

For each particle $\mathbf{s}^{\bullet}{}_{\bar{t},\ell}$, we execute C^{SITR} iterations of [100], (e.g., $C^{\text{SITR}} = 10$), to effectively discriminate non-homogeneous foregrounds from the backgrounds. We draw N^{SP} particles (e.g., $N^{\text{SP}} = 100$) equally around each of the object positions in the most recent $N^{\bullet}$ frames (e.g., $N^{\bullet} = 16$). Let $\mathcal{M}$ be the foreground mask returned by the segmentation method [100]. For each particle $\mathbf{s}^{\bullet}{}_{\bar{t},\ell}$, we extract color histogram $\boldsymbol{\Theta}_{\bar{t},\ell}$ only within the area defined by the foreground mask $\mathcal{M}$. Then, we use χ^2 kernel for defining our observation likelihood

$$p(\mathbf{z}^{\bullet}{}_{\bar{t}}|\mathbf{s}^{\bullet}{}_{\bar{t},\ell}) = \exp\left(-\frac{\parallel \boldsymbol{\Theta}_{\bar{t},\ell} - \bar{\boldsymbol{\Theta}} \parallel^2}{\left|\boldsymbol{\Theta}_{\bar{t},\ell} - \bar{\boldsymbol{\Theta}}\right|^1}\right), \tag{50}$$

where $\|\cdot\|^2$ and $\|\cdot\|^1$ are l_2 and l_1 norms, respectively, and $\bar{\boldsymbol{\Theta}}$ is the color histogram of the target reference $\bar{\mathbf{s}}^{\bullet}$. We obtain the target reference $\bar{\mathbf{s}}^{\bullet}$ by searching for the BB with the minimal gradient within the positive SVMs pool retained in the MIST.

We can now estimate the optimal location and the size of the BB $\hat{\mathbf{s}}^{\bullet}_{\hat{t}}$ using (49) and use it to re-initialize the proposed MIST to recover from the current failure state. The proposed object segmentation integration with MIST is summarized in Algorithm 2.

Algorithm 2: Proposed object segmentation integration with object tracking.

```
Input  : Internal variables: $\phi(\mathbf{x},\hat{\mathbf{y}}),\ k(\mathbf{x},\bar{\mathbf{x}}),\ k^\star(\mathbf{x}^\star,\bar{\mathbf{x}}^\star),\ k^\diamond(\mathbf{x}^\diamond,\bar{\mathbf{x}}^\diamond),\ k^{\text{MI}}(\mathbf{u},\hat{\mathbf{u}})$ in $F_t$.
Output: Prediction of the segmented BB $\hat{\mathbf{s}}^\bullet_{\hat{t}}$ in $F_t$.
repeat
    Construct the feature vector $\mathbf{x}^\bullet$ using (39).
    Learn the binary SVM online using the first $\bar{N}^{\text{EX}}$ frames.
    Predict any tracking failures, i.e., $y^\bullet(\mathbf{x}^\bullet) = -1$, using (44).
    if $y^\bullet(\mathbf{x}^\bullet) = -1$ then
        Draw $N^{\text{SP}}$ particles around each of the object positions in the most recent $N^\bullet$
        frames.
        for each particle $\ell$ do
            Propagate particles according to (4).
            Evaluate the foreground mask $\mathcal{M}$ by a segmentation method [100].
            Evaluate observation likelihood $p(\mathbf{z}^\bullet{}_{\bar{t}}|\mathbf{s}^\bullet{}_{\bar{t},\ell})$ by (50).
            Update the importance weights $\omega_{\bar{t},\ell}$ using (48).
        Estimated the optimal particle state $\hat{\mathbf{s}}^\bullet_{\hat{t}}$ with (49).
        Re-initialize the proposed MIST with $\hat{\mathbf{s}}^\bullet_{\hat{t}}$.
until end of video sequence.
```

3.4 Conclusion

In this Chapter, we described the proposed MIST and how we improved it by integrating object segmentation. Effective object tracking requires modeling target dynamics and appearance changes. Leveraging harmonic means and particle filter to formulate target dynamics improves accuracy and efficiency of object localization. Structured SVMs possess good generalization ability with built-in flexibility to model object appearances, while being effective in the presence of estimation noise. Accuracy of object tracking based on Structured SVMs is improved when the following are effectively explored: *1*) adaptive joint kernels using orthogonal features for Structured SVMs learning and inference, *2*) motion-augmented regularization for constraining the output search space, and *3*) conditional model updates for structured maximization problem.

Tracking failure is inevitable due to challenging factors inherent in videos, such as deformation, illumination changes, occlusion etc. Therefore, effective tracking also requires both tracking failure detection and re-initialization after its failure. Integrating object segmentation with tracking minimizes tracking failures and improves overall accuracy of object tracking.

Tracking failure can be detected using online binary SVMs framework. Recent history of the object trajectory provides rich clues for relocating lost target, for which particle filter is effective to sample and estimate the target state and object segmentation provides accurate modeling of the likelihood.

The selection of the segmentation method to integrate in case of tracking failure clearly affects the overall tracking results. On the other side, a different object tracking method may differently benefit from the integration of the same segmentation method. We observe that accuracy and speed of the overall tracking strongly depends on the three steps: segmentation method selected, object tracking method selected, and tracking failure method used.

Chapter 4

Experimental Results

4.1 Overview

In this Chapter, first, we discuss the experimental setup. We use large datasets and classify the video sequences into several challenging categories. Second, we present experimental results of the proposed MIST and objectively and subjectively compare it with several state-of-the-art trackers. Third, we present the results of the proposed integration of MIST with an active contour based object segmentation method using several challenging video sequences. Finally, we conclude the Chapter.

4.2 Experimental Setup

For our experiments, we used 50 video sequences (total $29,490$ frames, with frame size ranging from 250×350 to 800×1000) from the [7] benchmark dataset. Following [7], we divide the 50 test videos into 11 challenging categories to evaluate the effectiveness of the trackers according to: fast motion, occlusion, illumination variation, deformation, background clutter, low resolution, scale variation, in-plane rotation, out-plane rotation, out-of-view, and motion blur. We compare our method against 10 state-of-the-art object tracking algorithms: *Struck* [66], *ASLA* [121], *SCM* [87], *TLD* [65], *MIL* [64], *CT* [85], *CSK* [122], *L1APG* [56], *Frag* [123], and *IVT* [53]. We used a PC with Intel i5 1.8 GHz CPU and 4 GB of RAM to execute the proposed method and the original implementation of the compared methods provided by the respective authors. We use the CPU implementation of *Struck* [66], since the implementations of our and other compared trackers are based on CPU. Our method involves random numbers, therefore we report the mean result of 5 executions on each video sequences.

We apply four widely used [85, 124–127] objective measures for evaluating the selected tracking algorithms: center location error, overlap score, precision plot, and success plot. The center location error is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. The second evaluation metric is the overlap score $\Delta(\mathbf{y}_i, \mathbf{y}) = 1 - \frac{\mathbb{A}(\mathbf{y}_i \cap \mathbf{y})}{\mathbb{A}(\mathbf{y}_i \cup \mathbf{y})}$ defined in Chapter 3. Notice that $\Delta(\mathbf{y}_i, \mathbf{y}) = 1$ means identical match between the candidate $\mathbf{y}$ and ground truth $\mathbf{y}_i$ BBs, and $\Delta(\mathbf{y}_i, \mathbf{y}) = 0$ means no similarity. The third and fourth measures are precision plot and success plot [7]. The precision plot is based on center location error metric while the success plot is based on the overlap metric. The precision plot shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. A success plot is computed by measuring the fraction of frames with overlap score (varied from 0 to 1) that is greater than a given threshold. Notice that by plotting the precision and success plots for all thresholds, no parameters are required, which makes the plots unambiguous and intuitive to interpret. A higher precision score at low center error thresholds and higher success score at high overlap thresholds mean the tracker is more accurate. As the representative precision score for each tracker, the threshold for the score is normally set to 20 pixels, and for success score, the representative threshold is typically set to 0.5 [128].

4.3 MIST Evaluation

In this Section, we list the parameters of our method, and present representative quantitative and qualitative tracking results. We then demonstrate some results of the internal analysis of the proposed algorithm, and discuss limitations of the proposed method.

4.3.1 MIST Parameters

The parameters of the proposed MIST are listed in Table 1. We used 10 video sequences (*carDark*, *david3*, *trellis*, *soccer*, *matrix*, *car4*, *sylvester*, *suv*, *jumping*, and *fleetface*) to experimentally determine optimal parameter values. The 10 videos were selected so that their attributes cover different challenging categories. We observed that our method is mainly sensitive to the penalty parameter C^{SVM}. In Section 4.3.4.3, we present the results of the experiments carried out to estimate C^{SVM}. Our experiments showed that increasing or decreasing the values of the parameters in Table 1 (except C^{SVM}) about $10\% - 25\%$ did not noticeably affect the accuracy of the proposed MIST.

Parameter	Description	Value
N^{HM}	Number of states in the KHM motion model	8
N^{P}	Number of particles set at the first frame in the motion model	1000
N^{PP}	Number of prior states considered for estimating number of particles	16
C^{SVM}	Structured SVM regularization parameter	25
C^{CMU}	SVM model update threshold	0.02
γ^{MI}	Precision parameter of motion GRBF	0.20
$\gamma^{\diamond}$	Precision parameter of color GRBF	0.22

Table 1: Parameters of the proposed MIST.

4.3.2 Quantitative Comparison of MIST

In Table 2, we list the averaged objective measures for all sequences as well as the average frame rates obtained for all videos. We also present these objective measures for individual sequences in Tables 3 and 4. As can be observed from these Tables, our method well outperforms the compared trackers in the overall objective measures. Compared to the 3 most accurate methods (*Struck*, *ASLA*, and *SCM*) of Tables 2, 3, and 4, the proposed method is the fast with 11.23 Frames Per Seconds (FPS), which can be improved by down-sampling the input frames, or skipping frames, or by implementing MIST on a Graphic Processing Unit (GPU).

Figure 2 depicts the averaged success and precision plots for all sequences, and confirms

Tracker	Mean overlap score	Mean center error	Frame rate (FPS)	Code
MIST [Ours]	**0.551**	**26.84**	11.23	*C*
Struck [66]	*0.481*	*50.81*	9.46	*C*
TLD [65]	0.432	52.27	24.12	*MC*
ASLA [121]	0.457	62.17	4.88	*MC*
CSK [122]	0.400	88.88	**230.72**	*M*
CT [85]	0.256	84.63	*44.95*	*MC*
IVT [53]	0.373	77.56	23.22	*MC*
L1APG [56]	0.360	73.39	1.23	*MC*
SCM [87]	0.437	64.35	0.36	*MC*
MIL [64]	0.354	61.74	21.12	*C*
Frag [123]	0.320	77.28	4.29	*C*

Table 2: Mean objective measures and average frame rates for 50 test video sequences. The Code column states which programming language each tracker is coded; C:C/C++, M: Matlab, MC: Matlab and C/C++.

that our method outperforms the other trackers in both measures. With Figures 3 and 4, we also report the effectiveness of the proposed tracker with others on various challenge attributes, such as fast motion, occlusion, background clutter, etc. Figures 3 and 4 illustrate that the our method strongly competes against state-of-the-art effectively handling the challenging situations, except in the scale variation category where its effectiveness is similar to those of *SCM* and *ASLA*.

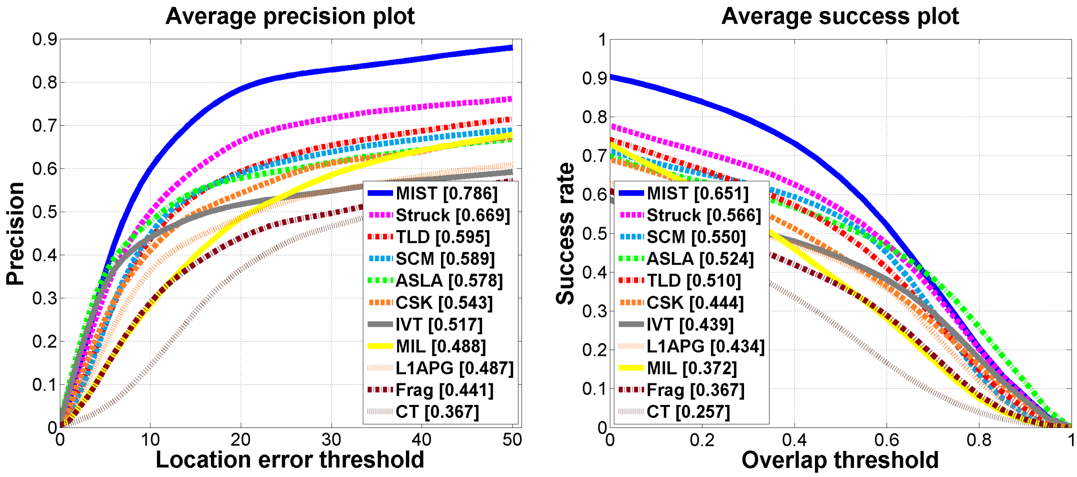

Figure 2: The averaged precision and success plots of the proposed MIST and the compared trackers on all 50 sequences.

	Tracker										
Sequence	*MIST*	*Struck*	*TLD*	*ASLA*	*CSK*	*CT*	*IVT*	*L1APG*	*SCM*	*MIL*	*Frag*
	[Ours]	[66]	[65]	[121]	[122]	[85]	[53]	[56]	[87]	[64]	[123]
carDark	1.4	1.4	27.5	*1.1*	3.8	28.7	8.4	**0.9**	3.4	45.7	79.7
david	13.1	63.0	5.1	**4.2**	17.7	11.6	*4.4*	70.7	9.6	21.1	99.3
trellis	**3.9**	27.1	31.1	31.8	18.8	46.6	125.8	62.2	*11.6*	68.9	56.3
soccer	**35.6**	75.6	77.1	86.0	70.1	82.9	146.4	101.3	158.1	*37.0*	127.4
matrix	**23.2**	195.4	57.2	56.8	113.7	59.2	144.8	57.1	52.5	*44.4*	184.5
ironman	**28.0**	116.2	*93.2*	93.9	185.5	165.3	132.3	173.0	167.4	189.2	252.5
deer	**3.8**	6.9	30.9	139.3	*4.8*	240.5	181.9	24.2	86.0	57.5	98.8
skating1	75.2	75.4	145.8	*48.9*	**7.8**	150.7	140.0	92.3	73.7	156.8	137.8
shaking	46.2	23.3	37.1	19.3	17.6	115.0	228.1	109.8	**13.9**	*14.5*	178.1
singer1	22.0	12.2	8.0	**3.0**	14.2	18.3	11.3	97.9	*4.3*	22.7	56.2
singer2	**12.2**	173.8	58.3	68.9	185.9	147.0	*15.4*	191.4	111.7	169.1	97.3
coke	23.6	**11.8**	25.1	59.7	*13.6*	31.6	82.3	101.5	28.5	48.3	207.0
bolt	390.7	386.2	**90.9**	367.9	430.3	*281.8*	389.5	402.1	432.4	387.2	333.8
boy	**3.3**	*3.4*	4.5	52.9	20.3	37.1	91.3	66.2	60.1	28.0	49.8
crossing	2.8	120.1	24.3	**1.5**	8.8	5.9	*2.3*	3.7	2.7	2.7	39.0
couple	*11.0*	30.0	**2.5**	57.6	145.2	77.3	123.9	31.8	157.3	34.3	36.6
football1	**6.8**	14.1	45.4	17.9	16.8	23.0	24.5	10.6	26.1	*8.0*	16.3
jogging-1	*18.4*	87.2	**6.7**	100.8	135.4	91.1	84.1	88.5	142.1	113.4	21.6
jogging-2	66.2	137.6	*13.6*	137.3	165.1	139.9	131.3	**5.6**	142.3	135.4	76.7
doll	**5.6**	11.7	*6.0*	17.1	44.8	16.3	15.2	114.8	7.4	21.7	11.8
girl	8.3	**2.8**	9.8	6.2	19.3	19.6	18.5	*3.7*	4.2	17.0	20.3
walking2	5.0	11.8	44.6	40.3	17.3	66.2	**2.9**	6.4	*3.1*	43.5	64.0
walking	7.3	5.3	10.2	*1.8*	6.7	434.4	**1.6**	3.1	3.6	4.5	9.8
david3	**10.1**	107.1	208.1	55.3	56.2	89.6	52.4	93.2	104.3	*33.7*	61.0
carScale	14.6	34.5	22.6	21.2	83.3	77.7	**11.9**	17.2	*12.2*	32.9	31.0
skiing	**5.4**	252.9	279.4	251.8	247.3	258.0	274.5	258.3	*242.6*	256.8	279.7
motorRolling	143.7	143.5	**80.9**	180.2	622.1	168.8	181.0	194.6	159.5	165.6	*141.5*
mountainBike	10.8	9.4	216.1	9.0	**6.5**	87.0	*7.7*	12.1	18.2	7.7	206.0
lemming	**14.0**	36.7	*16.0*	203.8	114.2	122.0	184.1	172.5	162.8	74.0	17.6
liquor	**27.4**	72.0	*37.6*	51.3	160.6	178.7	118.6	90.7	81.9	140.5	91.7
woman	12.7	**3.4**	139.9	156.0	207.1	121.6	196.0	133.5	*10.5*	124.2	103.8
faceocc1	52.1	19.2	27.4	97.7	**11.9**	25.7	*17.6*	22.7	20.2	34.9	19.2
basketball	23.0	85.4	213.9	249.8	**6.5**	96.5	117.4	114.2	232.4	106.4	*11.8*
subway	5.8	*3.3*	150.3	4.4	164.8	10.9	126.3	148.8	**2.2**	6.8	16.2
tiger1	*22.5*	**14.4**	49.5	92.3	70.2	83.5	106.6	64.3	81.1	35.4	55.9
tiger2	**15.8**	*19.1*	37.1	89.5	59.6	80.8	105.1	79.9	63.8	42.7	86.5
car4	27.9	4.2	86.2	**1.7**	19.5	85.2	*2.2*	101.4	8.4	53.8	147.6
sylvester	**5.3**	*6.3*	7.3	17.2	10.1	17.6	34.2	23.8	10.2	14.6	20.4
suv	**12.1**	36.2	*13.1*	73.1	573.2	86.3	57.3	91.4	74.9	73.4	41.2
jumping	**5.7**	7.0	*5.9*	39.9	85.7	45.0	61.6	33.5	41.0	13.3	7.4
fleetface	**16.4**	*20.3*	41.2	25.2	25.6	54.2	62.2	63.1	25.2	21.3	69.3
freeman1	11.5	11.4	39.7	13.1	125.7	14.5	11.6	*10.2*	**7.7**	12.2	11.2
freeman3	36.4	24.4	29.3	**2.5**	54.1	42.2	35.9	19.0	*6.8*	25.2	8.8
dog1	8.3	6.0	4.2	5.0	*3.9*	7.9	**3.5**	9.5	11.3	7.8	16.6
freeman4	**24.0**	43.6	39.2	60.8	78.7	95.0	43.0	*33.8*	98.0	76.7	42.8
football	20.5	13.8	14.3	*8.6*	16.0	15.6	14.3	17.6	**7.3**	12.5	16.5
faceocc2	10.6	*6.3*	12.3	20.1	**5.9**	26.0	7.4	10.9	11.4	16.9	39.8
fish	**3.1**	3.3	6.5	*3.3*	41.2	10.6	4.5	9.1	9.6	19.3	26.9
dudek	*10.4*	10.8	18.1	11.9	13.4	33.4	**9.7**	64.7	58.0	43.7	86.6
david2	1.8	*1.7*	5.0	10.1	2.3	59.6	**1.2**	25.8	9.8	16.2	15.7
mhyang	3.2	2.6	9.5	**1.7**	3.6	32.7	*1.9*	8.2	8.1	9.5	13.9
Mean	**26.8**	*50.8*	52.3	62.2	88.9	84.6	77.6	73.4	64.3	61.7	77.3
#Best score	20	4	4	6	5	0	6	2	4	0	0
#Second best score	4	7	7	5	3	1	7	3	7	5	2

Table 3: Comparison of MIST against 10 state-of-the-art trackers on the center error metric of the 50 video sequences. #Best and #Second best scores are the total number of sequences that each tracker performs best and second best on the center error metric, respectively.

	Tracker										
Sequence	*MIST*	*Struck*	*TLD*	*ASLA*	*CSK*	*CT*	*IVT*	*L1APG*	*SCM*	*MIL*	*Frag*
	[Ours]	[66]	[65]	[121]	[122]	[85]	[53]	[56]	[87]	[64]	[123]
carDark	*0.866*	0.863	0.449	0.827	0.716	0.117	0.663	**0.885**	0.730	0.153	0.073
david	0.501	0.228	*0.718*	**0.754**	0.402	0.464	0.679	0.247	0.625	0.373	0.087
trellis	*0.632*	0.434	0.484	0.619	0.480	0.278	0.277	0.200	**0.669**	0.264	0.315
soccer	**0.401**	0.151	0.127	0.140	0.145	0.143	0.151	0.168	0.108	*0.279*	0.169
matrix	**0.485**	0.101	0.156	*0.190*	0.031	0.155	0.022	0.177	0.180	0.117	0.015
ironman	**0.427**	0.098	0.102	*0.152*	0.119	0.082	0.049	0.081	0.109	0.050	0.028
deer	**0.756**	0.720	0.602	0.055	*0.748*	0.032	0.033	0.602	0.063	0.363	0.080
skating1	0.346	0.311	0.191	*0.482*	**0.497**	0.091	0.080	0.147	0.465	0.134	0.105
shaking	0.221	0.502	0.390	0.514	0.568	0.033	0.035	0.079	**0.612**	*0.581*	0.109
singer1	0.343	0.357	0.725	*0.776*	0.359	0.337	0.574	0.244	**0.830**	0.333	0.223
singer2	**0.669**	0.045	0.217	0.501	0.043	0.048	*0.569*	0.032	0.164	0.037	0.193
coke	0.452	**0.679**	0.396	0.166	*0.570*	0.336	0.119	0.095	0.424	0.243	0.032
bolt	0.017	*0.019*	**0.159**	0.011	0.019	0.007	0.010	0.009	0.008	0.011	0.015
boy	**0.775**	*0.768*	0.662	0.369	0.654	0.323	0.260	0.331	0.325	0.384	0.379
crossing	*0.742*	0.312	0.403	**0.806**	0.506	0.604	0.307	0.669	0.690	0.732	0.288
couple	0.474	0.492	**0.772**	0.210	0.075	0.195	0.074	0.483	0.064	*0.498*	0.448
football1	**0.607**	0.455	0.377	0.485	0.456	0.208	0.557	0.491	0.368	*0.560*	0.345
jogging-1	*0.577*	0.175	**0.770**	0.185	0.178	0.176	0.177	0.149	0.133	0.152	0.517
jogging-2	0.113	0.136	*0.656*	0.136	0.141	0.061	0.142	**0.736**	0.106	0.114	0.105
doll	0.548	0.540	0.570	**0.836**	0.316	0.479	0.497	0.075	*0.719*	0.345	0.494
girl	0.599	**0.741**	0.572	0.635	0.364	0.272	0.173	*0.692*	0.644	0.341	0.457
walking2	0.492	0.510	0.306	0.353	0.465	0.266	0.659	*0.697*	**0.748**	0.266	0.260
walking	0.566	0.552	0.446	*0.766*	0.537	0.003	**0.766**	0.730	0.649	0.535	0.479
david3	**0.734**	0.281	0.097	*0.551*	0.492	0.304	0.544	0.300	0.301	0.501	0.484
carScale	0.395	0.410	0.450	**0.656**	0.415	0.354	*0.626*	0.467	0.532	0.413	0.358
skiing	**0.493**	0.044	0.066	*0.096*	0.059	0.059	0.078	0.066	0.083	0.090	0.028
motorRolling	*0.165*	0.132	**0.229**	0.095	0.090	0.098	0.090	0.082	0.106	0.116	0.121
mountainBike	0.640	0.682	0.200	0.698	*0.716*	0.434	**0.726**	0.645	0.621	0.701	0.122
lemming	**0.654**	0.483	0.531	0.142	0.332	0.253	0.126	0.126	0.131	0.493	*0.568*
liquor	**0.696**	0.608	0.518	*0.638*	0.252	0.199	0.226	0.307	0.334	0.201	0.327
woman	*0.703*	**0.750**	0.133	0.150	0.191	0.102	0.148	0.146	0.549	0.154	0.135
faceocc1	0.474	0.718	0.585	0.249	**0.795**	0.619	*0.735*	0.632	0.662	0.537	0.675
basketball	0.548	0.428	0.022	0.080	**0.707**	0.165	0.085	0.173	0.078	0.229	*0.640*
subway	0.723	*0.750*	0.183	0.742	0.194	0.542	0.160	0.188	**0.816**	0.681	0.466
tiger1	*0.531*	**0.632**	0.376	0.182	0.259	0.105	0.095	0.249	0.117	0.387	0.333
tiger2	**0.626**	*0.562*	0.261	0.087	0.170	0.141	0.086	0.163	0.226	0.382	0.136
car4	0.407	0.491	0.206	**0.870**	0.465	0.215	*0.861*	0.243	0.752	0.253	0.099
sylvester	**0.742**	*0.722*	0.674	0.601	0.625	0.526	0.517	0.400	0.618	0.550	0.490
suv	**0.729**	0.475	*0.692*	0.466	0.524	0.166	0.406	0.400	0.455	0.246	0.549
jumping	*0.649*	0.596	**0.664**	0.223	0.050	0.059	0.122	0.274	0.133	0.404	0.601
fleetface	**0.657**	*0.635*	0.486	0.623	0.587	0.554	0.457	0.572	0.621	0.615	0.479
freeman1	0.390	0.364	0.280	*0.447*	0.236	0.305	0.426	0.360	**0.587**	0.293	0.349
freeman3	0.211	0.183	0.445	**0.752**	0.297	0.025	0.394	0.272	*0.502*	0.066	0.279
dog1	0.546	0.544	0.587	*0.702*	0.546	0.532	**0.741**	0.617	0.641	0.527	0.500
freeman4	*0.226*	**0.244**	0.224	0.147	0.123	0.014	0.149	0.157	0.123	0.039	0.189
football	0.579	0.532	0.489	0.568	0.560	0.451	0.557	0.553	**0.626**	*0.584*	0.507
faceocc2	0.706	**0.782**	0.616	0.585	*0.780*	0.476	0.727	0.702	0.690	0.643	0.492
fish	*0.866*	0.866	0.814	**0.876**	0.208	0.703	0.778	0.724	0.702	0.528	0.472
dudek	0.734	0.736	0.648	*0.753*	0.716	0.602	**0.775**	0.463	0.591	0.513	0.505
david2	*0.848*	**0.858**	0.693	0.469	0.825	0.032	0.702	0.332	0.506	0.376	0.605
mhyang	0.808	*0.818*	0.632	**0.909**	0.796	0.324	0.796	0.739	0.733	0.685	0.608
Mean	**0.551**	*0.481*	0.432	0.457	0.400	0.256	0.373	0.360	0.437	0.354	0.320
#Best score	15	7	5	8	3	0	4	2	7	0	0
#Second best score	11	7	3	11	4	0	4	2	2	5	2

Table 4: Comparison of MIST against 10 state-of-the-art trackers on the overlap ratio metric of the 50 video sequences. #Best and #Second best scores are the total number of videos that each tracker performs best and second best on the overlap ratio metric, respectively.

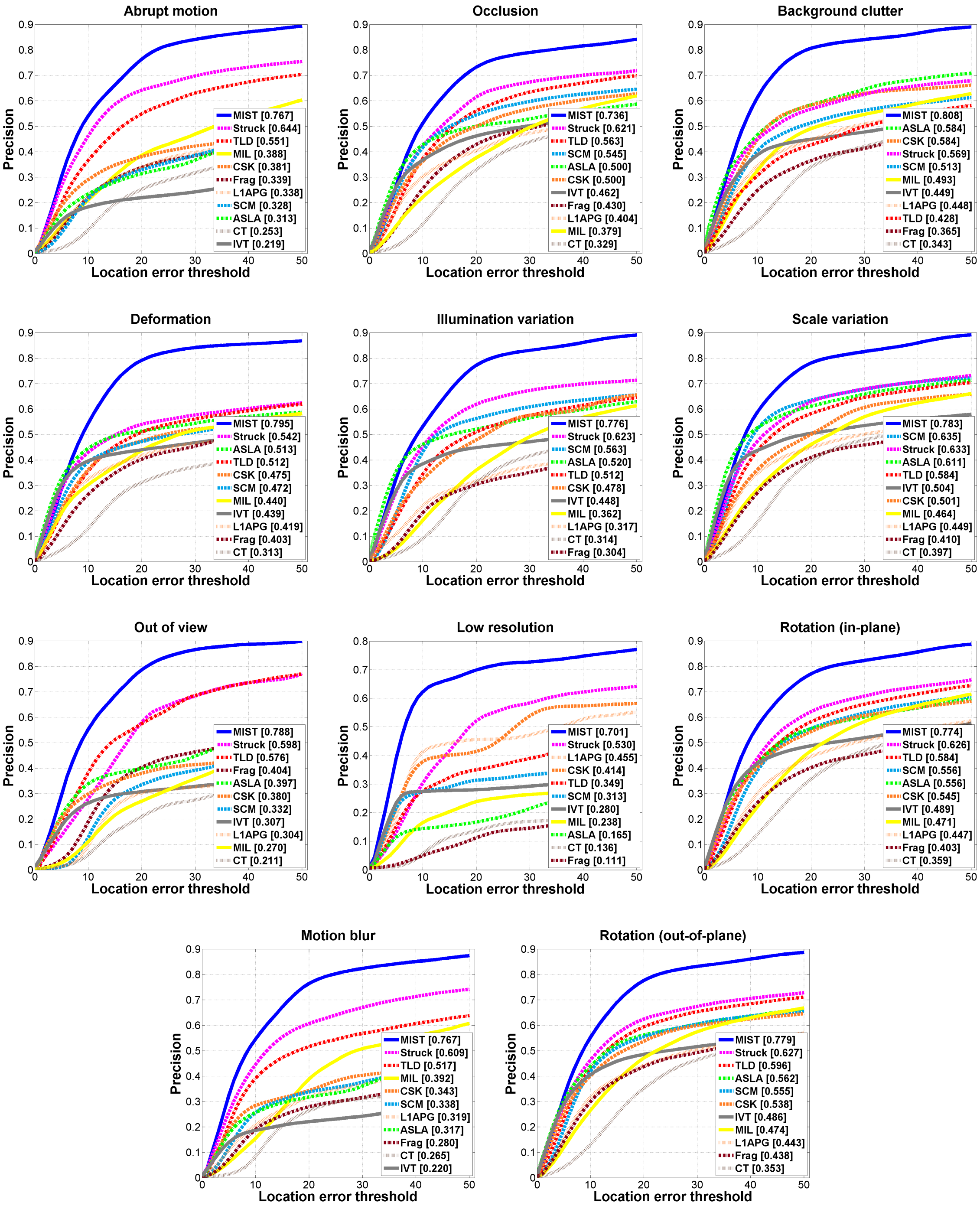

Figure 3: Precision plots of the proposed MIST and the compared trackers for 11 challenging categories on all 50 sequences.

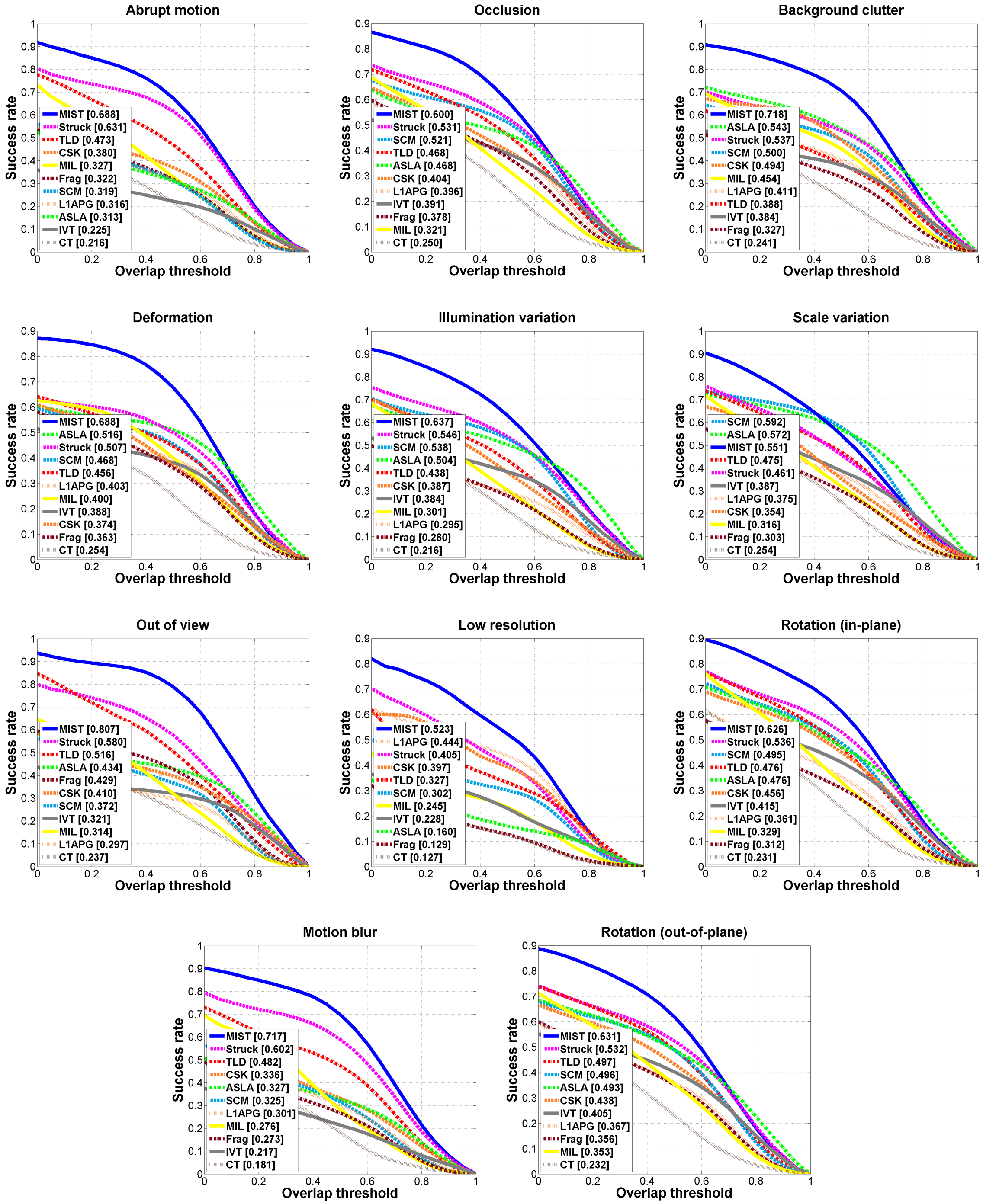

Figure 4: Success plots of the proposed MIST and the compared trackers for 11 challenging categories on all 50 sequences.

4.3.3 Subjective Comparison of MIST

Based on our objective experimental results above, we observe that *Struck*, *ASLA*, *SCM*, *TLD*, and *IVT* outperform the others. To retain the clarity in subjective figures, we present the subjective comparison results[1] of our method against these best performing trackers only.

4.3.3.1 Object deformation

We use *David3*, *David*, and *Crossing* sequences to evaluate the effectiveness of the trackers in handling object deformation. Figure 5 depicts some qualitative results of the compared trackers and the proposed method. In general, our method, *ASLA*, and *Struck* are more effective than *TLD*, *IVT*, and *SCM*. More specifically, in the *David3* sequence, the proposed method and *ASLA* (to a lesser degree) are more effective in handling drastic deformation and occlusion, while *Struck* gradually drifts away from the object around the middle of the sequence. *TLD* fails almost at the beginning of this sequence.

4.3.3.2 Fast motion

The *Boy*, *Jumping*, and *Dudek* video sequences are used to qualitatively evaluate the trackers in dealing with fast object motion. Some subjective results are shown in Figure 6. We observe that the proposed method, *Struck*, and *TLD* are more effective than *ASLA*, *IVT*, and *SCM*. In the *Boy* and *Jumping* sequences, the effectiveness of *ASLA* and *SCM* deteriorate during the abrupt object motion.

4.3.3.3 Occlusion

For evaluating the trackers against occlusion subjectively, we employ *Soccer*, *Liquor*, and *Matrix* video sequences. Some representative tracking results are shown in Figure 7 highlighting that the proposed method qualitatively outperforms the compared methods. Notice that, the object in this sequence also undergoes deformation and illumination changes.

[1] The subjective results are best viewed in color.

4.3.3.4 Illumination variation

We use *Basketball*, *Skiing*, and *Fish* sequences to subjectively evaluate the trackers effectiveness in handling illumination variation, and some results are depicted in Figure 8. We observe that our method qualitatively outperforms the others. More specifically, in the *Basketball* sequence, the object (player) is similar to background objects (players) in the scene, and the proposed method performs well while all other compared methods gradually lose tracking the object when it becomes closure to other objects in the background with similar colors. Furthermore, our method is very effective in tracking the miniature object throughout the *Skiing* sequence, while all other methods fail after the first few frames.

4.3.3.5 Background clutter

We use *Singer2*, *CarDark*, and *Football1* sequences to test the effectiveness of the respective trackers in dealing with background clutter. Figure 9 shows some qualitative results, and we observer that the effectiveness of the proposed method and *ASLA* (to a lesser degree) is better compared with all other tested trackers. More specifically, the object in the *Singer2* sequence is surrounded by significant background clutter, however, only the proposed method is able to track the object by effectively discriminating the background clutter throughout the sequence.

4.3.3.6 In-plane rotation

We use *Coke*, *Freeman4*, and *David2* video sequences to subjectively compare the trackers in dealing with in-plane rotation. Figure 10 depicts some qualitative results. In general, the proposed method, *Struck*, and *TLD* trackers are more effective than the other methods. More specifically, we observe that *ASLA* incorrectly learns its appearance model few frames after the beginning of the sequence and it fails to recover and track the object.

4.3.3.7 Out-plane rotation

Figure 11 depicts some results of the compared trackers and the proposed method in handling with out-plane rotation in *Trellis*, *Sylvester*, and *Jogging-1* vedeo sequences . In particular, *Struck*, *IVT* and *ASLA* trackers are suboptimal in tracking the object in these sequences, while

TLD is relatively effective. In the *Jogging-1* sequence, *Struck*, *IVT*, *ASLA*, and *SCM* drift away from the object when collusion occurs, while *TLD* is more effective. In contrast, the proposed tracker succeeds tracking the object throughout the sequences.

4.3.3.8 Low resolution

We employ *Walking2*, *Deer*, and *Ironman* sequence to evaluate the effectiveness of the trackers in dealing with scenarios with low contrast between the target and background, and some qualitative results are shown in Figure 13. We observe that the proposed method, *Struck*, and *TLD* are more effective than the other methods. In particular, in the *Deer* sequence, *ASLA*, *TLD*, *SCM*, and *IVT* get distracted by background clutter, drift away from the target gradually.

4.3.3.9 Scale variation

Figure 12 shows some qualitative results of which the object undergoes significant scale variations in *Doll*, *CarScale*, and *Couple* video sequences. Our method compared with *SCM* and *ASLA* keeps engaged with the object throughout the sequences avoiding tracking drifts. Our method is based on objects with fixed scale and therefore, its effectiveness can be further improved by adaptively computing the scale of the object.

4.3.3.10 Out-of-view

We use *Suv*, *Lemming*, and *Tiger2* sequences for evaluating the tracks when the object is out-of-view, and some qualitative results are shown in Figure 14. In general, the proposed method, *Struck*, and *TLD* are more effective than the other methods. In the *Suv* sequence, the proposed method and *TLD* successfully re-detect and continue tracking the object amidst its absence from the scene (around 550 frame).

4.3.3.11 Motion blur

For subjective evaluation of the trackers against motion blur, we use *Woman*, *Tiger1*, and *FleetFace* video sequences. Some qualitative results are shown in Figure 15, and we observe that the proposed method, *Struck*, and *SCM* perform better than other methods. In particular, we observer that, in the *Woman* sequence, both *TLD* and *ASLA* drift away from the object around the middle of the sequence, while the proposed method, *Struck*, and *SCM* keep tracking the object throughout the sequence.

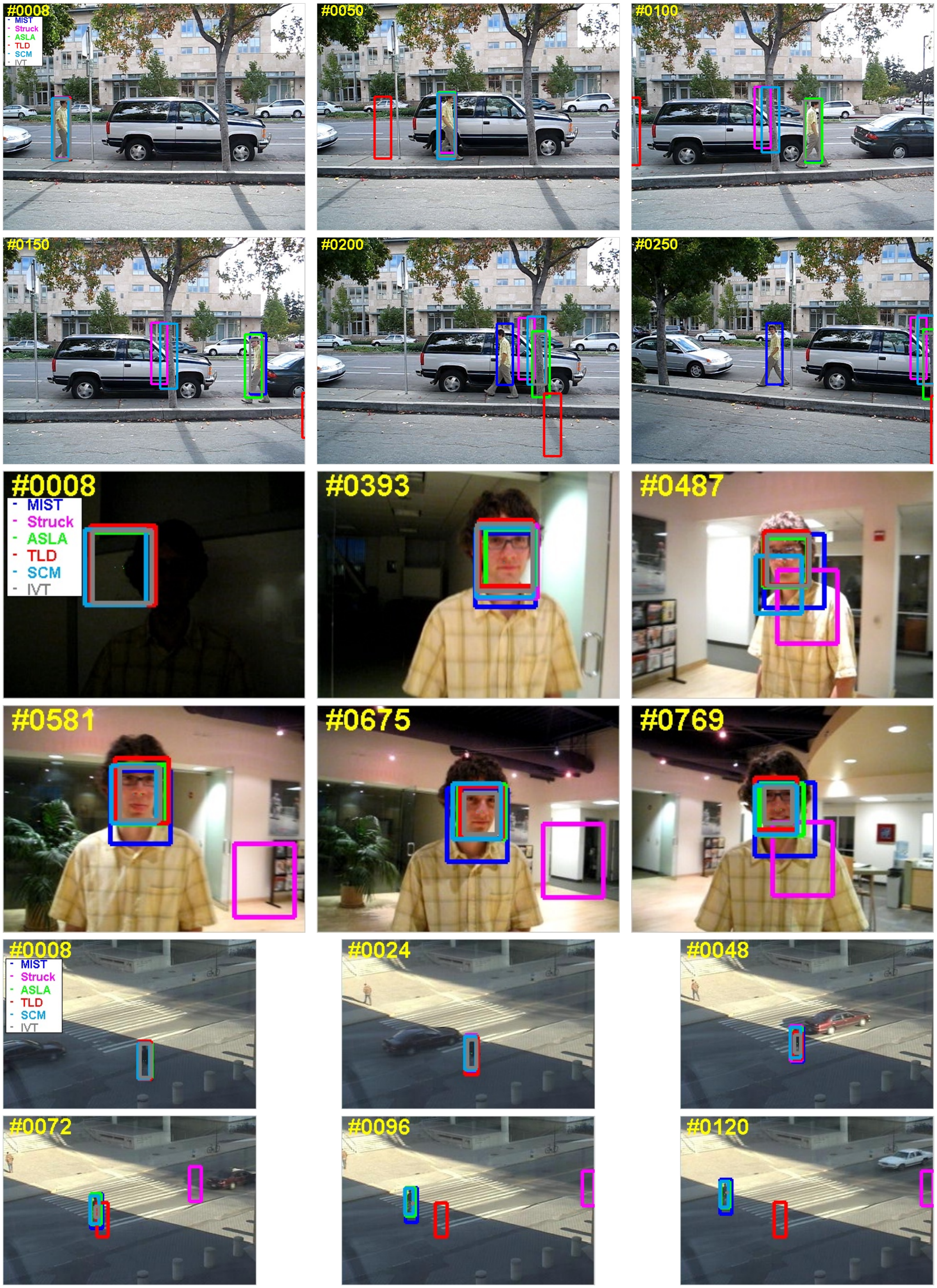

Figure 5: Deformation category: top *David3*; middle *David*; and bottom *Crossing*.

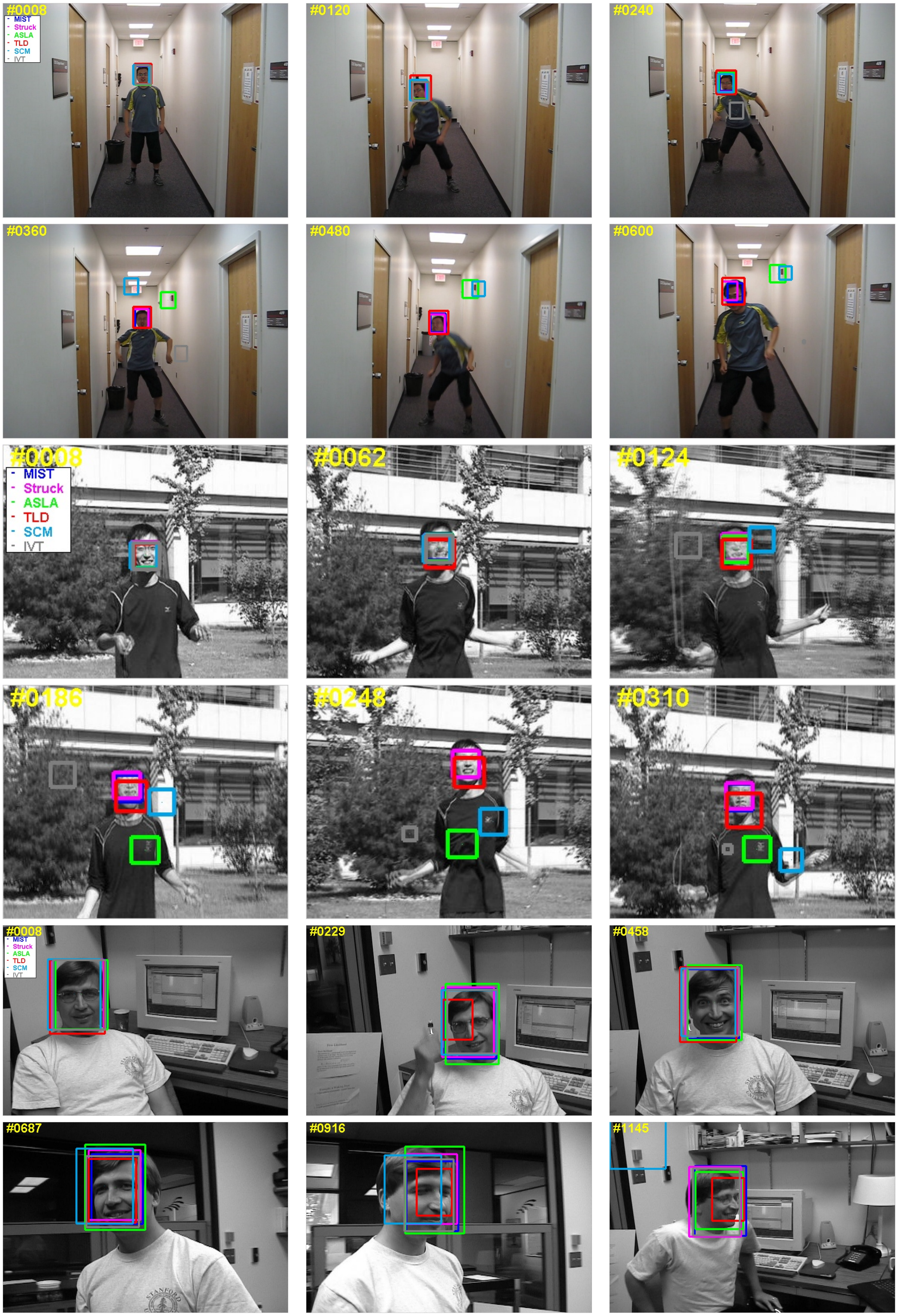

Figure 6: Fast motion category: top *Boy*; middle *Jumping*; and bottom *Dudek*.

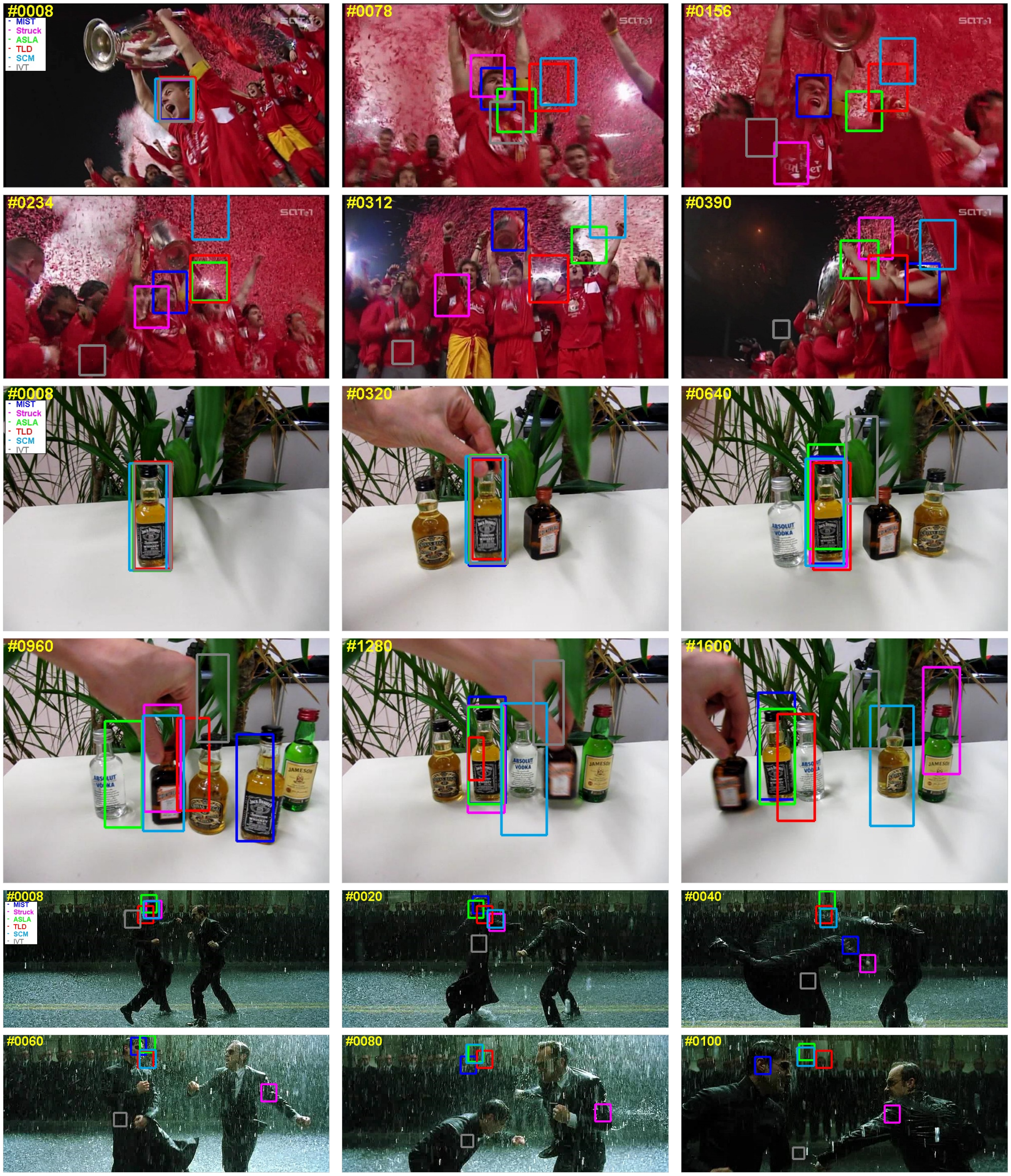

Figure 7: Occlusion category: top *Soccer*; middle *Liquor*; and bottom *Matrix*.

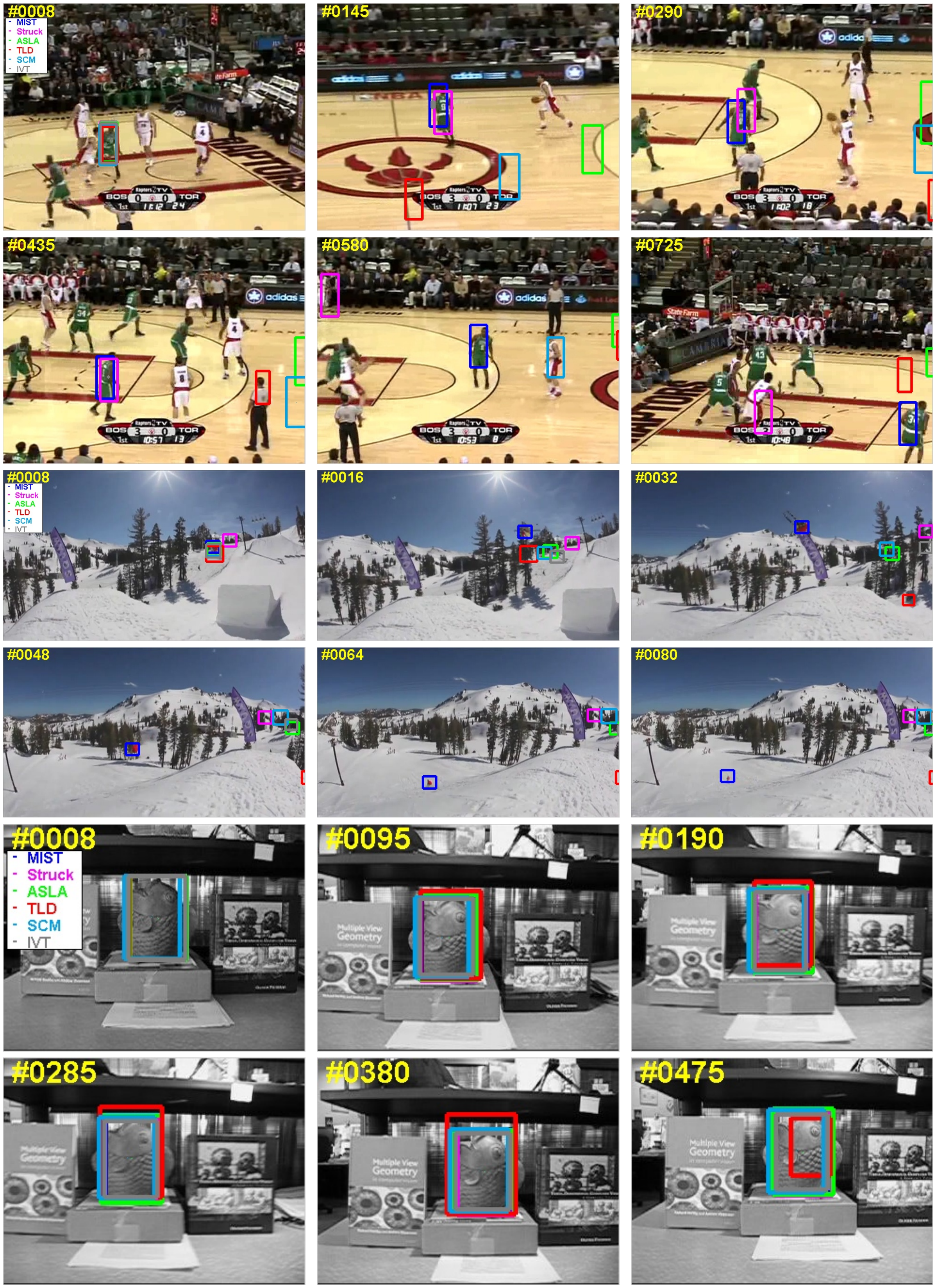

Figure 8: Illumination variation category: top *Basketball*; middle *Skiing*; and bottom *Fish*.

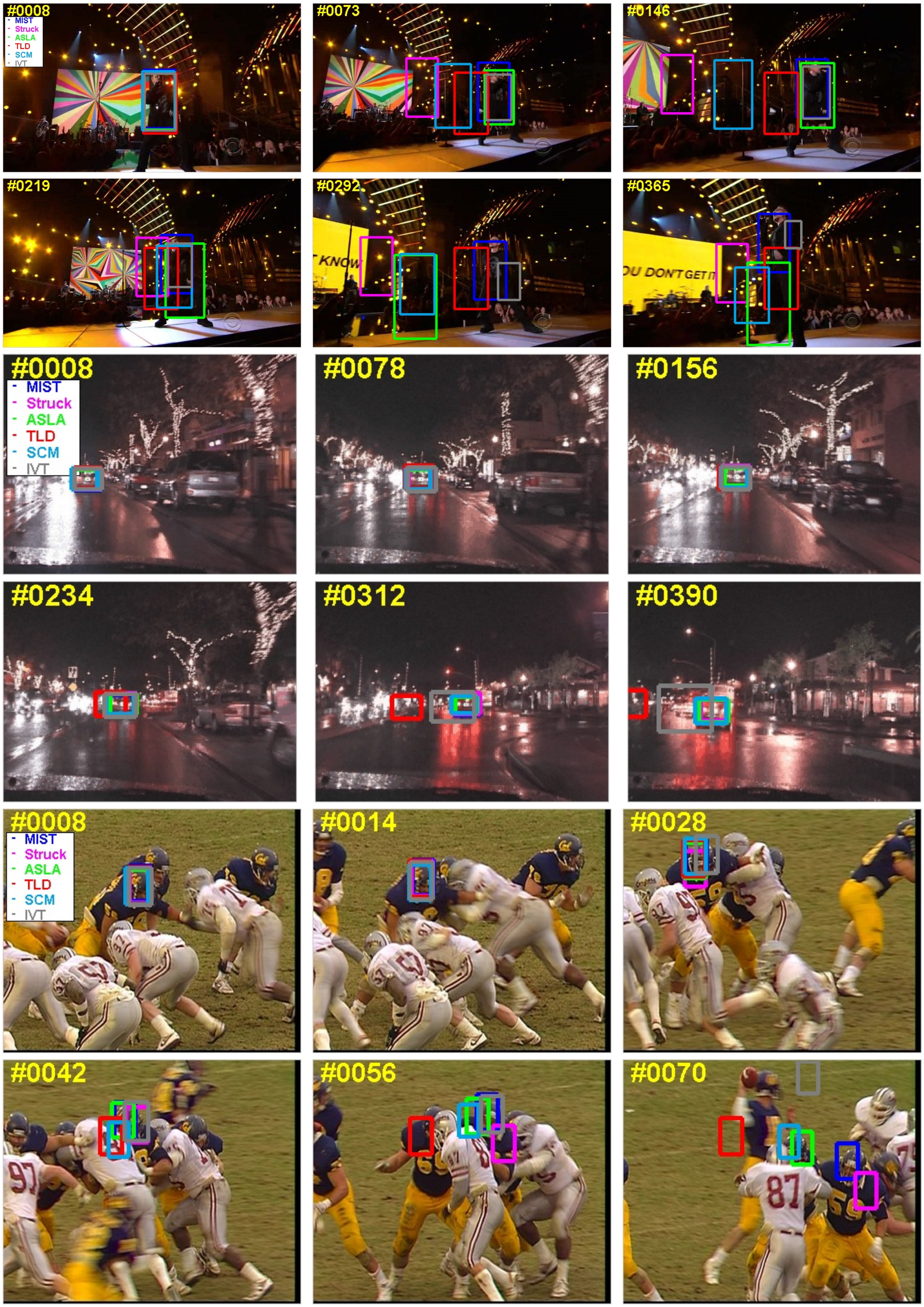

Figure 9: Background clutter category: top *Singer2*; middle *CarDark*; and bottom *Football1*.

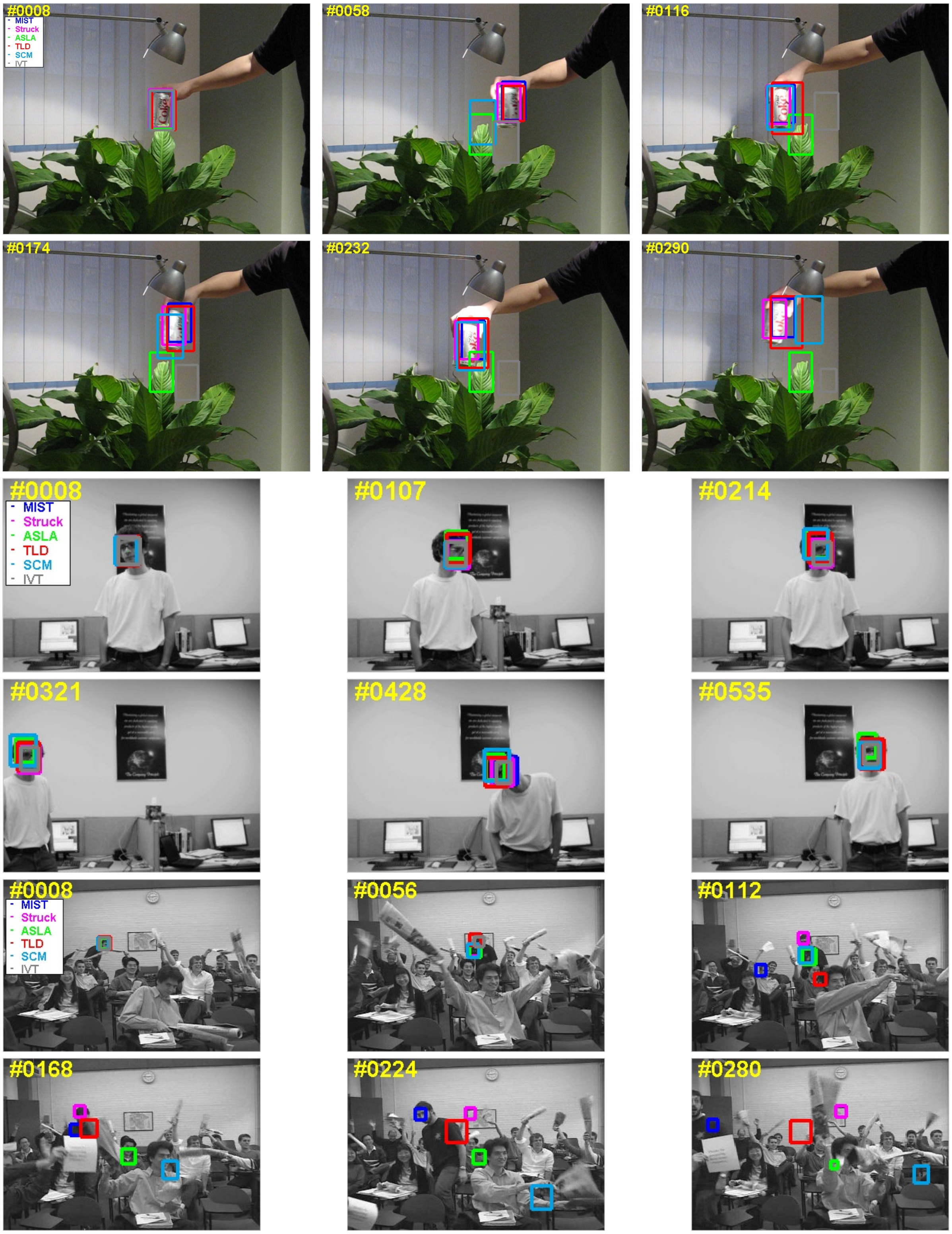

Figure 10: In-plane rotation category: top *Coke*; middle *Freeman4*; and bottom *David2*.

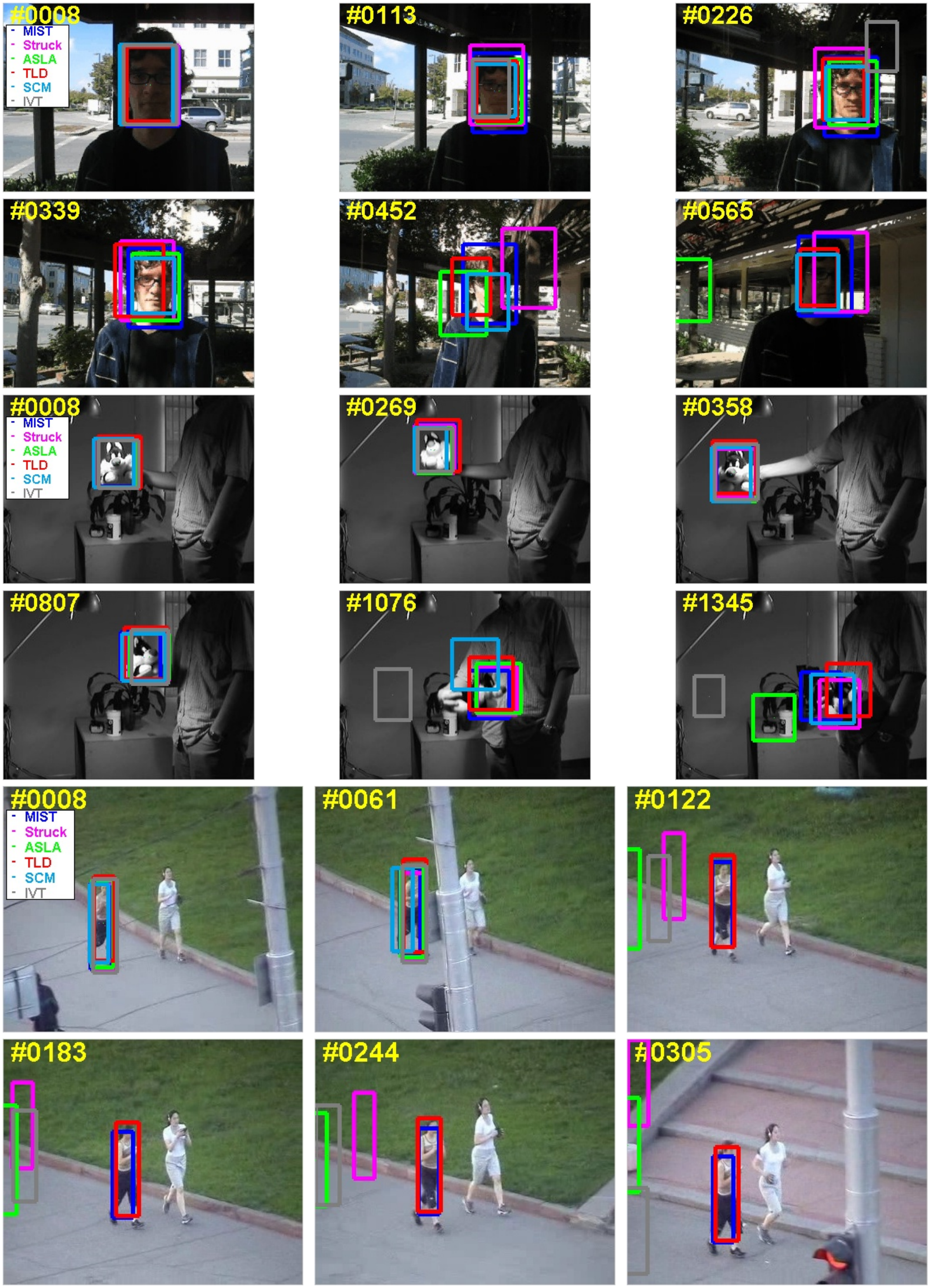

Figure 11: Out-plane rotation category: top *Trellis*; middle *Sylvester*; and bottom *Jogging-1*.

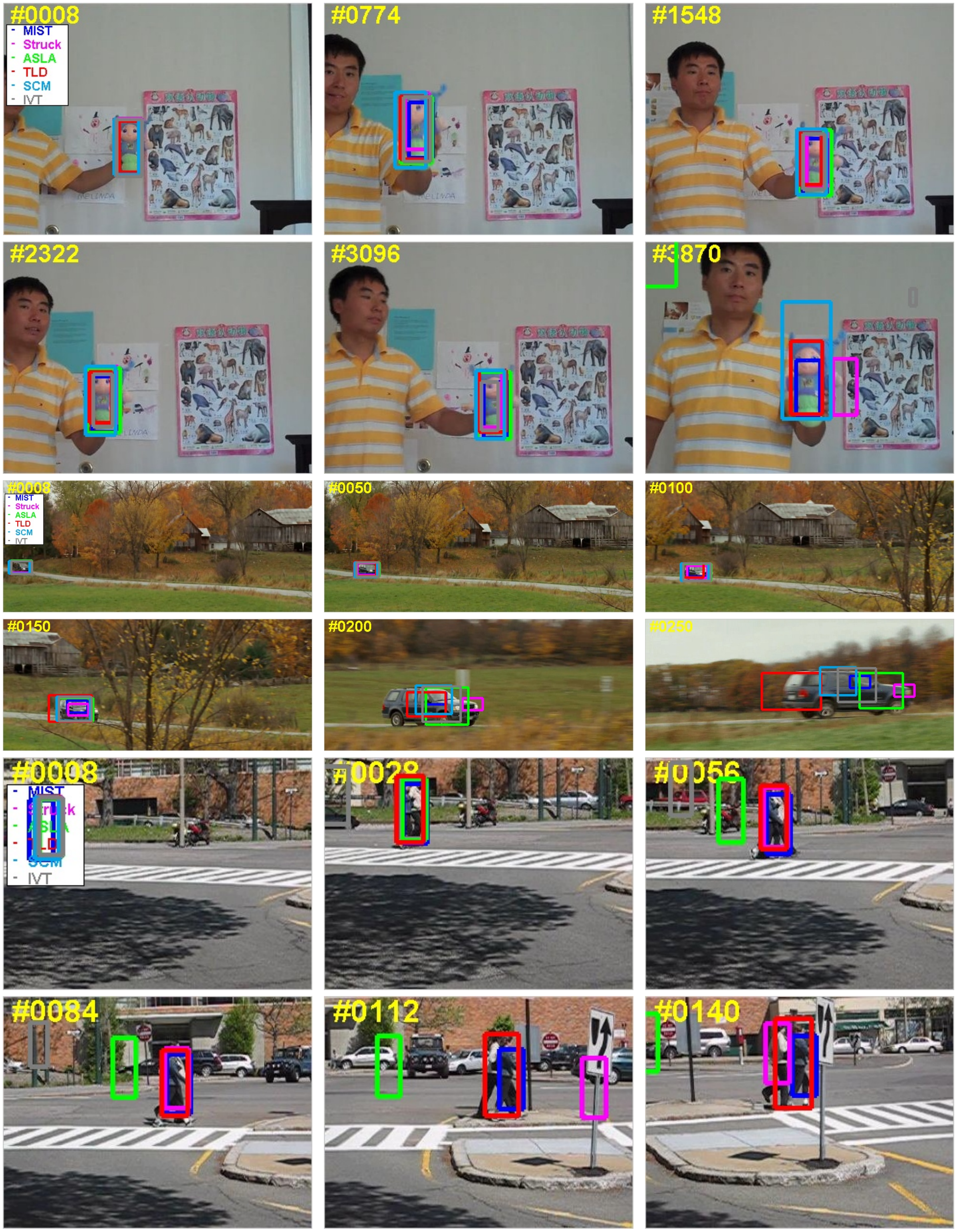

Figure 12: Scale variation category: top *Doll*; middle *CarScale*; and bottom *Couple*.

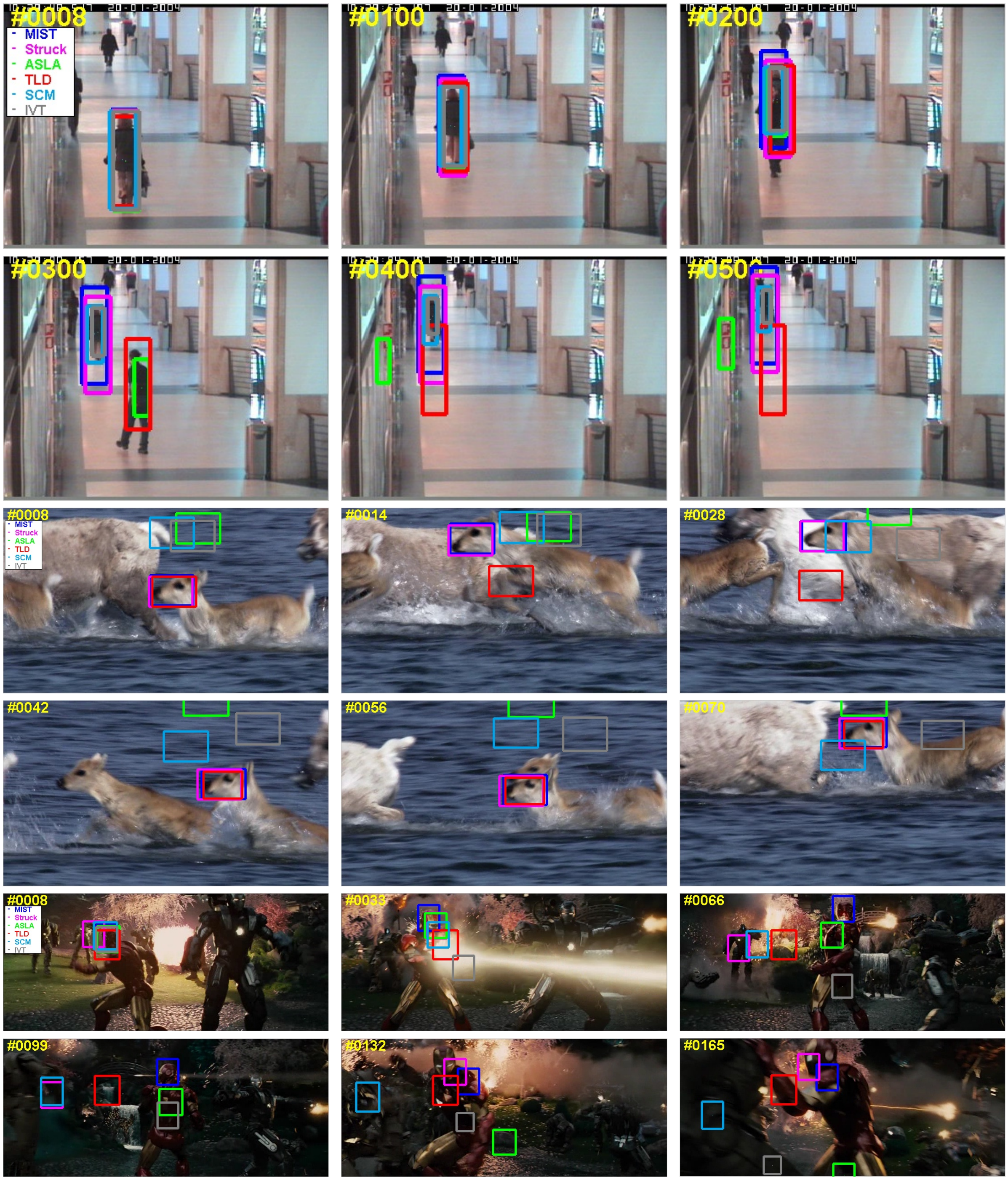

Figure 13: Low resolution category: top *Walking2*; middle *Deer*; and bottom *Ironman*.

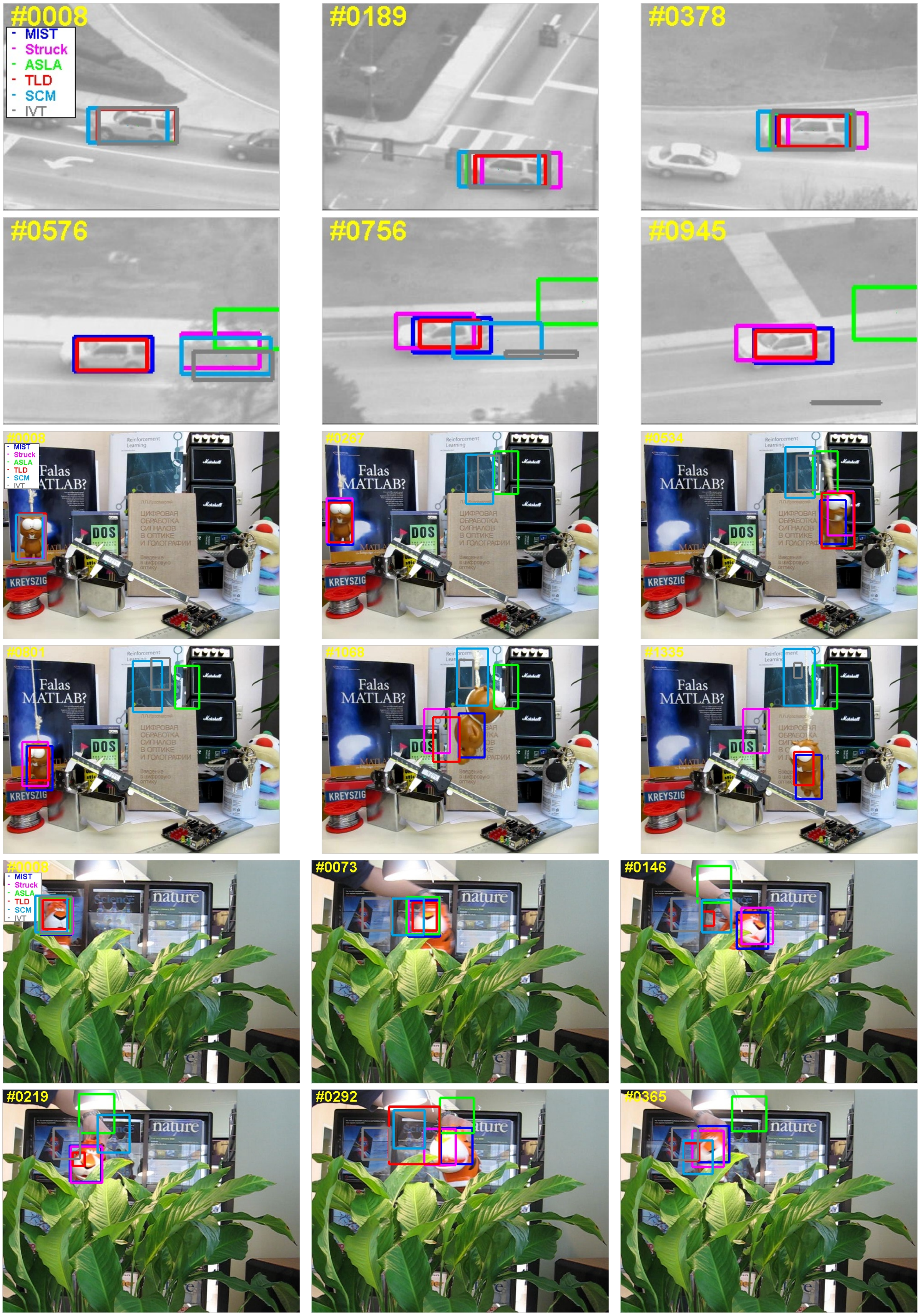

Figure 14: Out of view category: top *Suv*; middle *Lemming*; and bottom *Tiger2*.

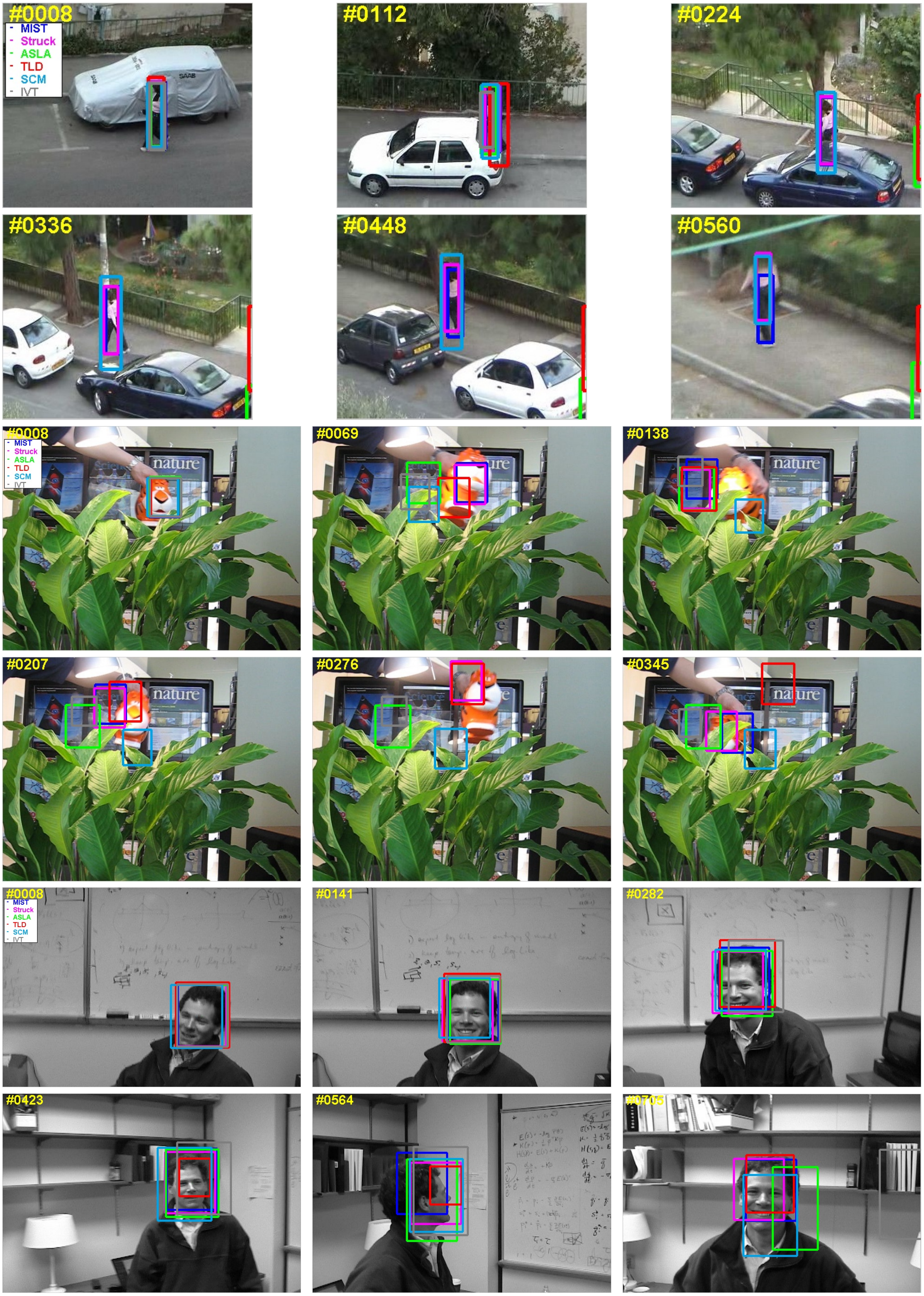

Figure 15: Motion blur category: top *Woman*; middle *Tiger1*; and bottom *FleetFace*.

4.3.4 Internal Analysis

In this Section, we present some results of the internal analysis of the proposed algorithm. We used the same 10 test video sequences used in Section 4.3.1 (i.e., *carDark*, *david3*, *trellis*, *soccer*, *matrix*, *car4*, *sylvester*, *suv*, *jumping*, and *fleetface*) to study the contribution of several components in the proposed tracker, and to experimentally determine optimal feature-kernel combinations and the parameter C^{SVM}.

4.3.4.1 Contribution of internal components

We implemented several derivatives of our tracker to investigate the contribution of several components. To that end, the effectiveness of color (MIST-CLR), adaptive learning rate in the JKF (MIST-BKG), Conditional Model Update (CMU) (MIST-CMU), and the employed HoG feature descriptor (MIST(Haar); i.e., Haar vs HoG) are studied by removing each components from the main algorithm. As can be seen from Figures 16, 17, and 18, our main algorithm MIST outperforms all other variations on all 11 challenging categories.

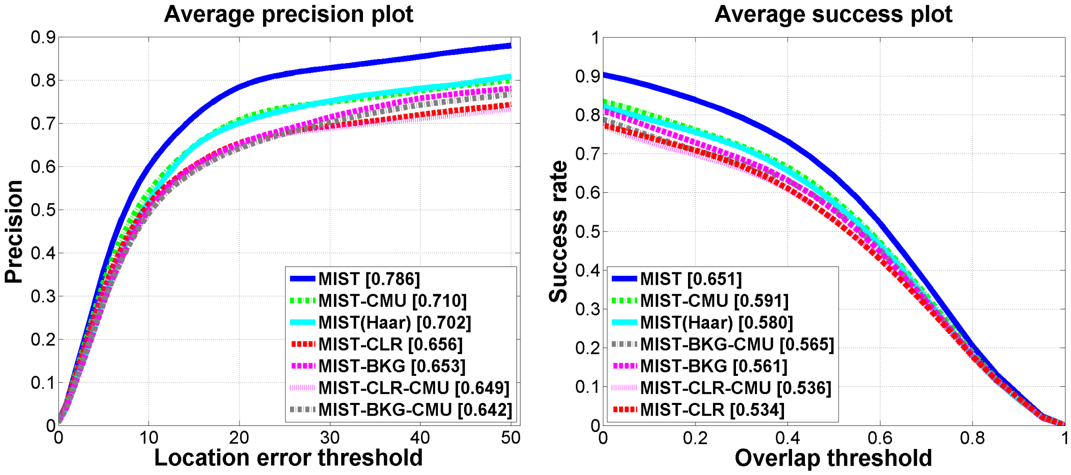

Figure 16: The averaged precision and success plots on all sequences for internal comparison.

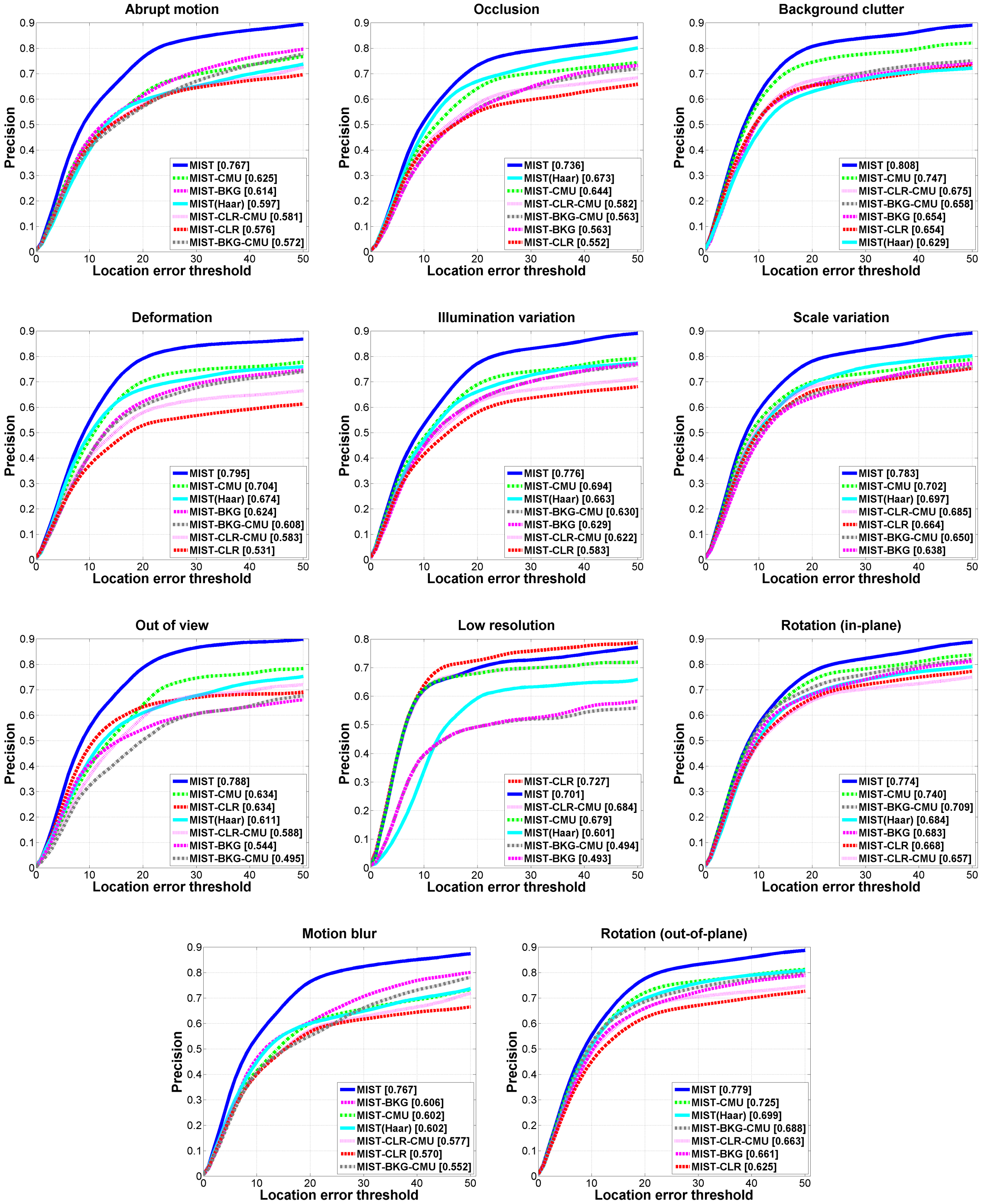

Figure 17: Precision plots for 11 challenging categories for internal comparison.

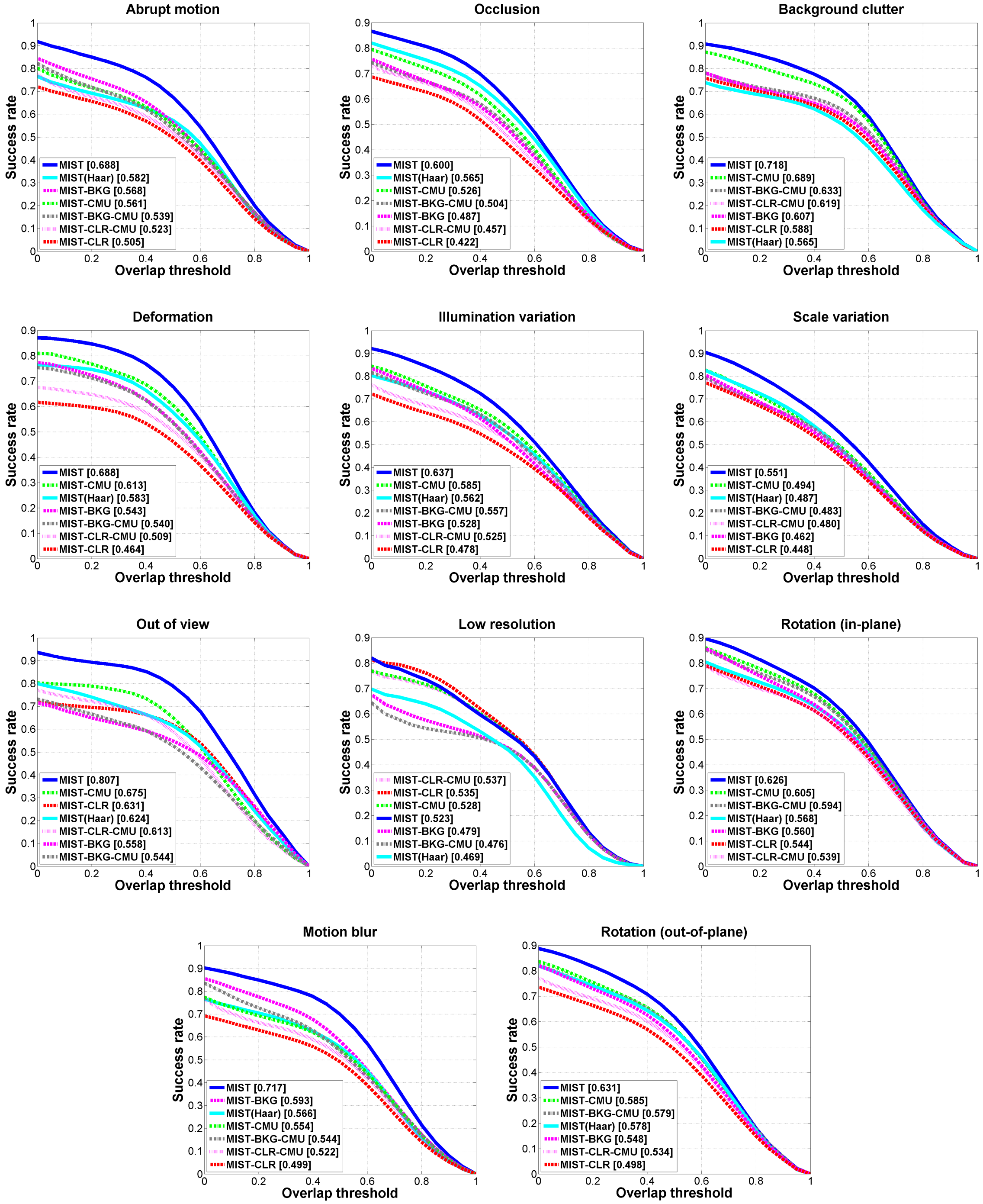

Figure 18: Success plots for 11 challenging categories for internal comparison.

4.3.4.2 Kernel design

For our joint kernel design, we consider using HSV color histogram [129] to measure the color similarity and HoG [111] as feature descriptor to encode global shape. Figure 19 shows the response of these features to the histogram-intersection (HistInt), GRBF, Bhattacharyya (BTCH), and χ^2 kernels. Based on the optimal overlap ratio, we, therefore, select the feature-kernel combinations: color with Bhattacharyya kernel and HoG with GRBF kernel. Note that we have also experimented other feature-kernel combinations, for example, raw pixels, Haar, Oriented FAST and Rotated BRIEF (ORB) were combined with kernels, such as linear, polynomial, and exponential; however, the responses of such feature-kernel combinations were weaker compared to those depicted in Figure 19.

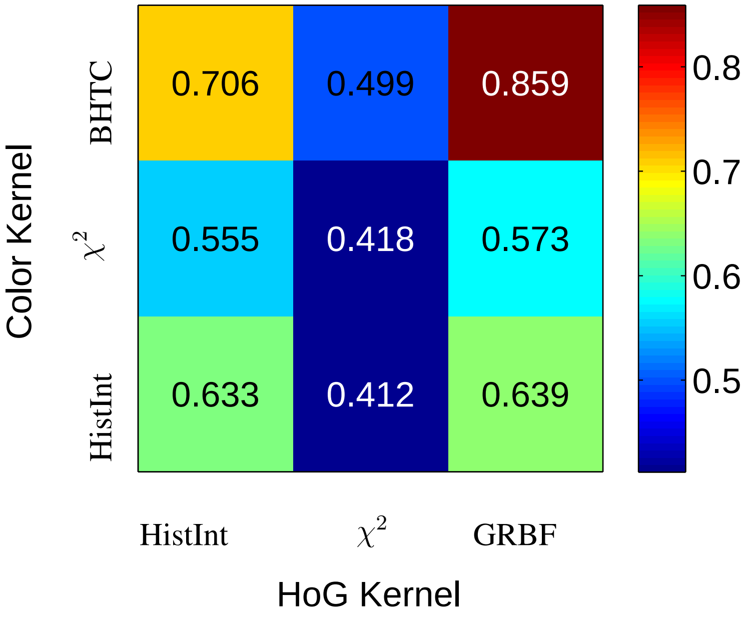

Figure 19: Confusion matrix of kernels and feature descriptors.

4.3.4.3 Penalty parameter C^{SVM}

The penalty parameter C^{SVM} in (20) determines the trade-off between the effectiveness and efficiency of the model, therefore tuning C^{SVM} is inevitable [52]. To select the optimal value for C^{SVM} for object tracking, we studied its influence by taking BB overlap ratio as effectiveness and frame rate as efficiency measures. Figure 20 shows respective overlap ratio and frame rate and their corresponding error bars. We calculate the standard deviation of the overlap ratio and frame rate for each parameter C^{SVM} in Figure 20. The each error bar in Figure 20 has a distance of 1 standard deviation above and below the respective plots. As can be seen from Figure 20, smaller C^{SVM} deviates the effectiveness while larger C^{SVM} penalizes the efficiency. With larger C^{SVM}, the SVM optimization chooses a hyperplane with a smaller margin and, therefore, the classification of the positive and negative samples is more accurate. Figure 20 shows that when $C^{\text{SVM}} \gtrsim 25$ the effectiveness reaches its optimum. For the proposed tracker, we thus set $C^{\text{SVM}} = 25$.

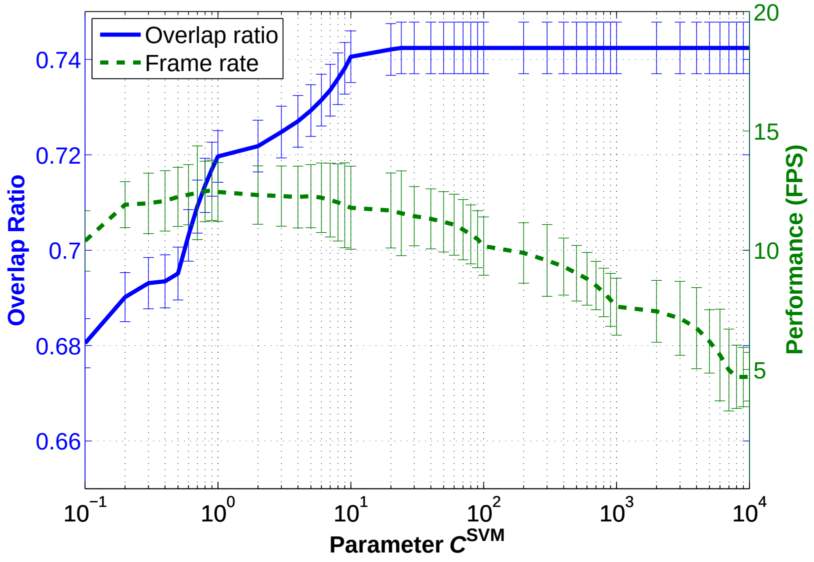

Figure 20: Influence of SVM parameter C^{SVM} on the efficiency and effectiveness (horizontal-axis is in $\log$ scale).

4.3.5 Limitations

Figure 21 depicts an inaccuracy case of our method. We observe that the implicit occlusion detection built within our conditional model update scheme is suboptimal in handling severe occlusion. This can be because, the employed HoG feature descriptors are suboptimal in encoding object contrast enough to discriminate it from the background. The effectiveness of the proposed method can be improved by incorporating Haar-like features in a Multiple Kernel Learning (MKL) [130] framework, for example.

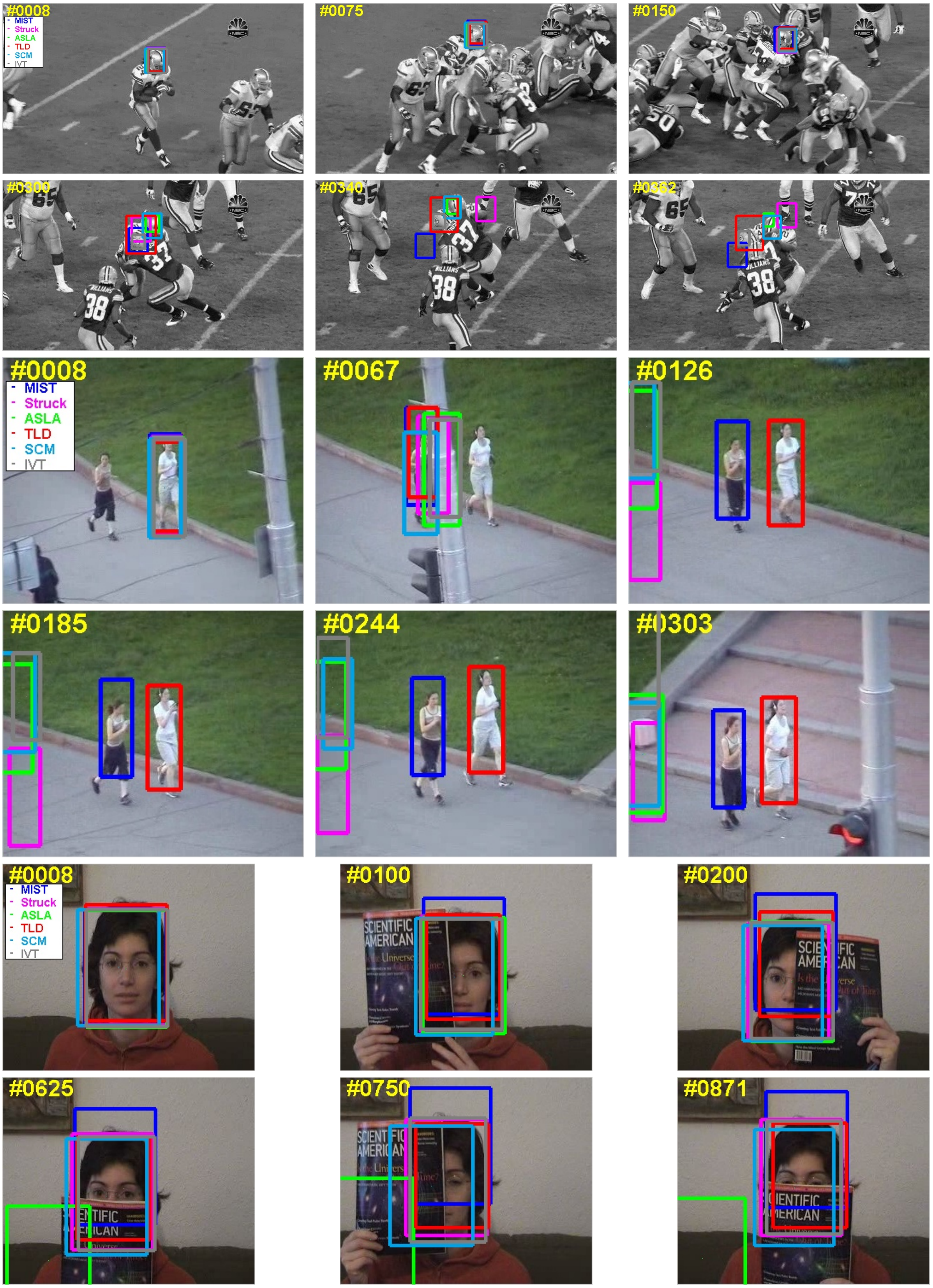

Figure 21: Inaccuracy cases of our method: the target in *football*, *jogging-2*, and *faceocc1* sequences undergoes severe occlusion.

4.4 MIST Integrated with Object Segmentation

In this Section, we present experimental results of the proposed object tracking method when integrated with object segmentation: MIST Integrated with Segmentation (MIST-SEG). First, we list the parameters used in the proposed MIST-SEG. Second, we study the effect on our MIST-SEG when integrated with several segmentation algorithms. Third, we present some quantitative and qualitative results, and finally we discuss limitations of the MIST-SEG.

Note that, our goal is to use segmentation to recover from tracking failures anywhere in the video sequence; thus, we do not run segmentation every frame, but rather only when we detect a tracking failure. This also implies that we do not use segmentation at the start of the sequence. Experimentally, we found that using segmentation to refine the initial (first frame) BB does not improve the overall tracking results, particularly on challenging video sequences. This means that the results of the proposed MIST and the proposed MIST-SEG are identical until the first failure is occurred.

4.4.1 Parameters of MIST Integrated with Object Segmentation

The parameters of the proposed technique of integrating segmentation with MIST are listed in Table 5. We empirically obtained optimal values of these parameters by testing the MIST-SEG on the same 10 test sequences (i.e., *carDark*, *david3*, *trellis*, *soccer*, *matrix*, *car4*, *sylvester*, *suv*, *jumping*, and *fleetface*) used for evaluating the parameters of the proposed MIST. These videos were selected from different challenging categories. Experimentally, we observed that our MIST-SEG is mainly sensitive to the number of iterations C^{SITR} that the segmentation is executed. With higher C^{SITR}, the employed method [100] removes more smooth areas of the object through its energy minimization, and with smaller C^{SITR}, it retains more background regions; in both cases, the method [100] is suboptimal in segmenting relevant foreground object from the background. The proposed MIST-SEG is not sensitive to variations of about 10% of the value of parameters in Table 5 that have smaller values, such as $\gamma^{\bullet}$. Varying about 20% of parameters with higher values, for example $N^{\bullet}$ or N^{SP}, also does not noticeably affect the effectiveness of the proposed MIST-SEG.

Parameter	Description	Value
$\bar{C}^{\text{SVM}}$	Binary SVM regularization parameter	25
$\gamma^{\bullet}$	Precision parameter of tracking failure-detection GRBF	1
$N^{\bullet}$	Number of most recent frames used in segmentation particle filter	16
N^{SP}	Number of particles in segmentation particle filter	100
$\bar{N}^{\text{EX}}$	Number of examples pairs in the binary Support-Vector-Machines	32
C^{SITR}	Number of segmentation iterations	10

Table 5: Parameters of the proposed MIST integrated with segmentation.

4.4.2 Comparison of Segmentation Methods Integrated with MIST

We manually integrate the proposed MIST with three different segmentation methods: active contour-based method [100], Lazy snapping method [118], K-means segmentation method [119]. The proposed integrated methods are denoted by MIST-SEG-AC, MIST-SEG-LS, and MIST-SEG-KM, respectively. We used the same test sequences in Section 4.4.1 to observe the effect on the proposed MIST when integrated with these segmentation methods.

In Table 6, we list the averaged overlap scores, center errors, and frame rates over all 10 video sequences. As can be seen from Table 6, MIST-SEG-AC outperforms the MIST-SEG-LS, MIST-SEG-KM, and MIST on the averaged center error and overlap score metrics, and MIST-SEG-AC is faster than MIST-SEG-LS and MIST-SEG-KM. In Tables 7 and 8, we present the center errors and overlap scores for each of the 10 video sequences. With regard to the total number of sequences that each of the tracker performs best and second best, we note that MIST-SEG-AC is best on overlap ratio metric (cf. Table 8) while MIST is slightly better than MIST-SEG-AC on the center error metric (cf. Table 7). The quantitative results in these Tables also confirm the effectiveness of the MIST-SEG-AC when compared with MIST-SEG-LS, MIST-SEG-KM, and MIST. Figures 22, 23, and 24 depict success and precision plots. We observe from these Figures that MIST-SEG-AC outperforms MIST-SEG-LS, MIST-SEG-KM, and MIST. Consequently, we have selected the active contour-based segmentation method [100] due to its effectiveness and efficiency. The segmentation method [100] localizes region-based active contour energies. We execute C^{SITR} iterations of the [100], (e.g., $C^{\text{SITR}} =$

10), to effectively discriminate non-homogeneous foregrounds from background. In the next Section, we present some results of manual (supervised) as well as automatic (unsupervised) integration of [100] with the proposed MIST, which we denote by MIST-SEGM and MIST-SEGA, respectively.

Tracker	Mean overlap score	Mean center error	Frame rate (FPS)
MIST-SEG-AC	**0.670**	**10.46**	*11.53*
MIST-SEG-LS	0.457	34.00	10.14
MIST-SEG-KM	0.326	63.14	10.56
MIST	*0.634*	*14.03*	**12.52**

Table 6: Mean objective measures and average frame rates of manually integrating the proposed MIST with [100], [118], and [119] on the 10 test video sequences.

	Tracker			
Sequence	*MIST-SEG-AC*	*MIST-SEG-LS*	*MIST-SEG-KM*	*MIST*
carDark	**1.3**	42.4	70.5	*1.4*
david3	11.2	*10.7*	238.8	**10.1**
trellis	*4.2*	7.5	12.9	**3.9**
soccer	**13.7**	80.5	65.9	*35.6*
matrix	**10.6**	55.6	78.4	*23.2*
ironman	**21.6**	100.0	67.8	*28.0*
deer	*3.9*	7.3	8.7	**3.8**
suv	14.1	**10.5**	33.2	*12.1*
jumping	**4.3**	*5.4*	33.7	5.7
fleetface	*19.5*	20.1	21.5	**16.4**
Mean	**10.5**	34.0	63.1	*14.0*
#Best score	5	1	0	4
#Second best score	3	2	0	5

Table 7: Individual center errors of the proposed MIST integrated with [100], [118], and [119].

4.4.3 Quantitative Comparison of MIST Integrated with Object Segmentation

In Table 9, we list the averaged objective measures and the averaged frame rates obtained for 50 video sequences. In Table 10, we present these objective measures for the 50 sequences

	Tracker			
Sequence	*MIST-SEG-AC*	*MIST-SEG-LS*	*MIST-SEG-KM*	*MIST*
carDark	**0.870**	0.090	0.004	*0.866*
david3	*0.718*	0.625	0.011	**0.734**
trellis	**0.655**	0.504	0.410	*0.632*
soccer	**0.582**	0.334	0.130	*0.401*
matrix	**0.608**	0.296	0.305	*0.485*
ironman	**0.492**	0.107	0.223	*0.427*
deer	**0.784**	0.749	0.698	*0.756*
suv	*0.654*	0.647	0.518	**0.729**
jumping	**0.738**	*0.654*	0.412	0.649
fleetface	*0.595*	0.562	0.551	**0.657**
Mean	**0.670**	0.457	0.326	*0.634*
#Best score	7	0	0	3
#Second best score	3	1	0	6

Table 8: Individual overlap ratios of the proposed MIST integrated with [100], [118], and [119].

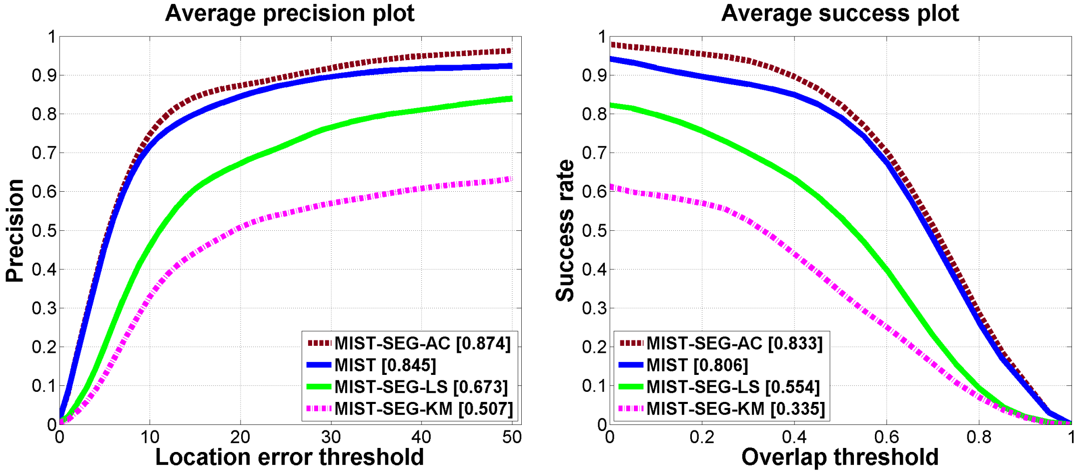

Figure 22: The averaged precision and success plots on all 10 sequences for comparing the proposed MIST manually integrated with [100], [118], and [119].

individually. As can be observed from the quantitative results in Tables 9 and 10, the integration of segmentation when tracking failure occurs improve the overall effectiveness of the proposed MIST. When MIST-SEG-AC is compared with MIST with respect to the total number of sequences that each of the tracker performs best and second best (in Table 9), MIST-SEG-AC performs best on both overlap ratio and center error metrics. Notice that in Tables 3 and 4, we list these scores for MIST and 10 other trackers. Consequently, the total

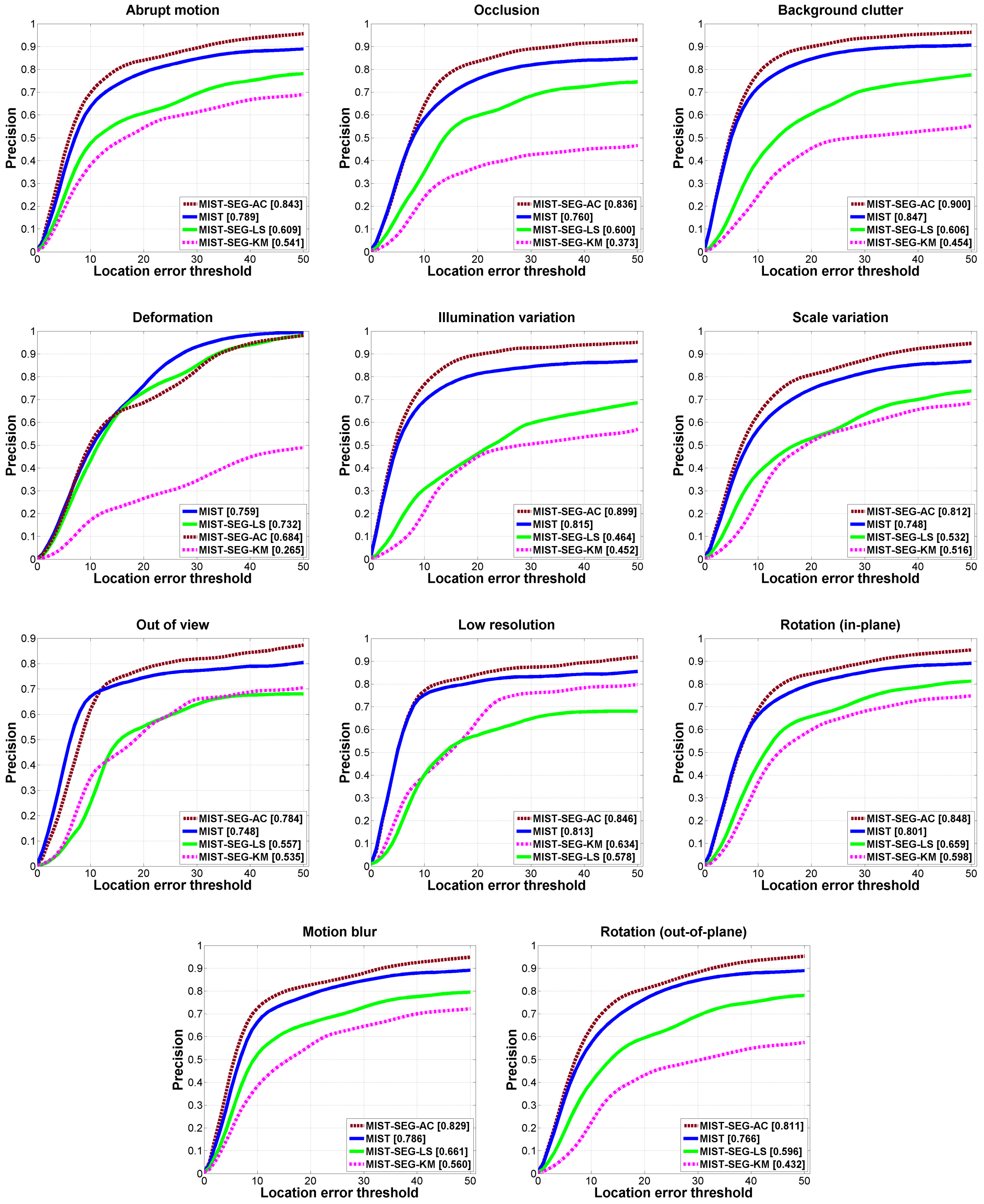

Figure 23: Precision plots for 11 challenging categories on all 10 sequences for comparing the proposed MIST manually integrated with [100], [118], and [119].

number of video sequences that MIST performs best and second best when it is compared with 2 trackers (MIST-SEGA and MIST-SEGM)) are different from those in Tables 3 and 4. Clearly, the manual segmentation (MIST-SEGM) performs better than the proposed automatic

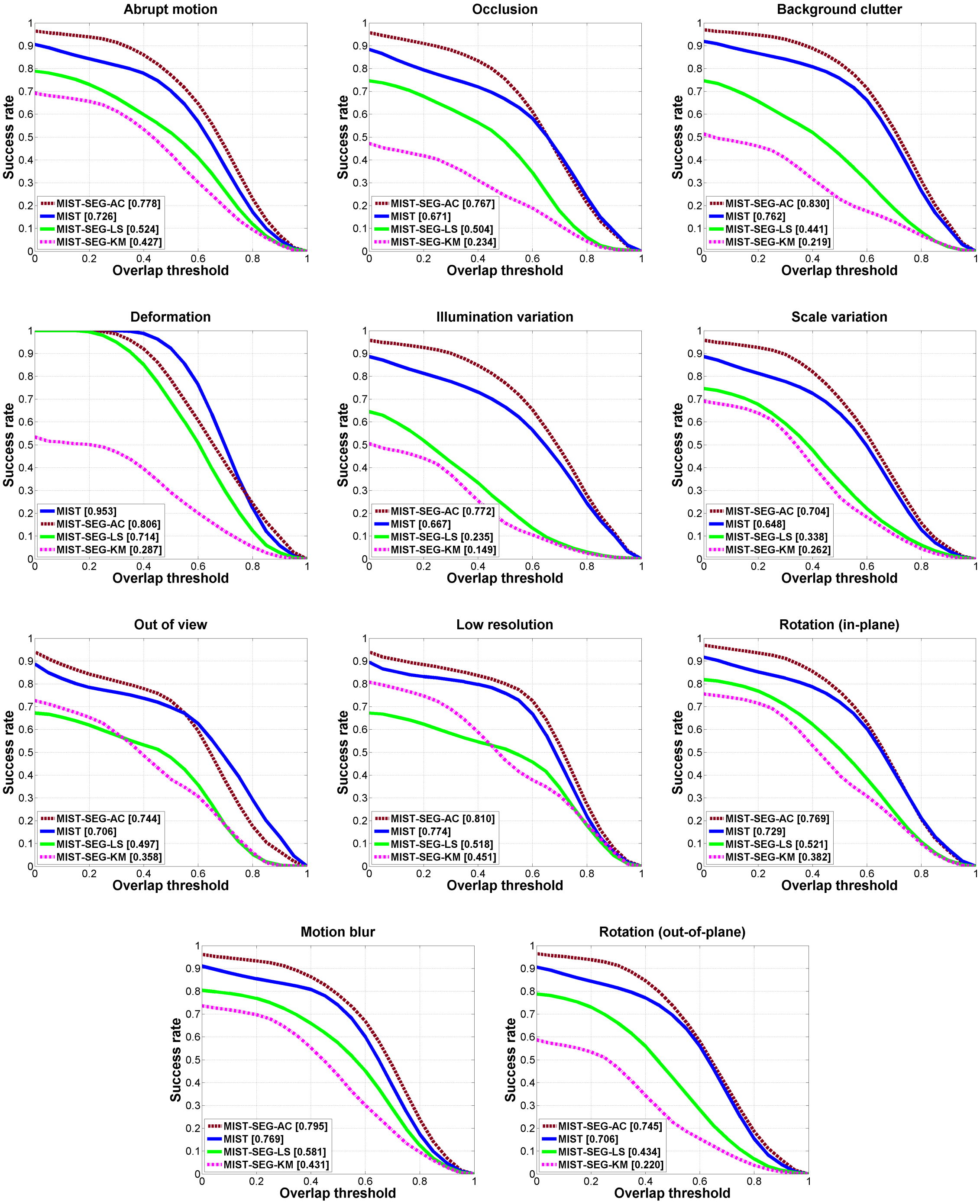

Figure 24: Success plots for 11 challenging categories on all 10 sequences for comparing the proposed MIST manually integrated with [100], [118], and [119].

segmentation (MIST-SEGA) in most cases. This is because the interactive selection of the object for segmentation is more accurate than its detection by the proposed (automatic) technique. The proposed MIST-SEGA is based on the observation of the particle filter which is derived from optimal BB among all the positive support vectors. During a tracking failure, the BBs of negative support vectors can be incorrectly labeled as positive due to drift, hence the observation to the particle filter can become suboptimal. Consequently, the overall effectiveness of MIST-SEGA can be worse than MIST-SEGM. Figures 25, 26, and 27 depict results based on precision and success metrics, which also affirm the effectiveness of object segmentation integration with object tracking. The results in these Figures further confirm that MIST-SEGM performs better than MIST-SEGA. In Table 11, we objectively evaluate the proposed tracking failure detection using the 10 test video sequences. In each video sequence, we count the number of tracking failures detected by our method, and we list the average number of failures detected for the 10 test video sequences in Table 11. The proposed failure detection method estimates the optimal location and the size of the BB $\hat{\mathrm{s}}^{\bullet}_{\hat{t}}$ using (49), which is subsequently used to re-initialize tracking. Using the ground truth data of the 10 test video sequences and the estimated BB $\hat{\mathrm{s}}^{\bullet}_{\hat{t}}$, we measure the overlap ratio and center error at each failure state. We present the mean of these objective measures in Table 11. As can be seen from Table 11, our method has a higher mean overlap ratio (closer to 0.5) and a lower mean center error, and therefore, our failure detection technique is reliable.

Tracker	Mean overlap score	Mean center error	Frame rate (FPS)
MIST-SEGA	*0.595*	*23.96*	10.58
MIST-SEGM	**0.619**	**18.29**	7.44
MIST	0.551	26.84	11.23

Table 9: Mean objective measures and average frame rates of the proposed automatic and manual integration of segmentation with MIST for 50 test video sequences.

4.4.4 Subjective Comparison of Manual and Automatic Integration of Object Segmentation with MIST

In this Section, we present qualitative results of the proposed MIST-SEGM and MIST-SEGA, and subjectively compare them against the proposed MIST.

Sequence	**Tracker**					
	Center Error			**Overlap Ratio**		
	MIST-SEGA	*MIST-SEGM*	*MIST*	*MIST-SEGA*	*MIST-SEGM*	*MIST*
carDark	1.5	**1.3**	*1.4*	0.862	**0.870**	*0.866*
david	*30.2*	33.0	**13.1**	*0.273*	0.239	**0.501**
trellis	**3.9**	4.2	*3.9*	*0.653*	**0.655**	0.632
soccer	*24.0*	**13.7**	35.6	*0.436*	**0.582**	0.401
matrix	27.4	**10.6**	*23.2*	*0.521*	**0.608**	0.485
ironman	**19.2**	*21.6*	28.0	**0.503**	*0.492*	0.427
deer	4.1	*3.9*	**3.8**	*0.756*	**0.784**	0.756
skating1	**10.5**	*18.2*	75.2	**0.640**	*0.597*	0.346
shaking	*12.5*	**11.0**	46.2	*0.647*	**0.694**	0.221
singer1	*13.4*	**12.6**	22.0	*0.572*	**0.609**	0.343
singer2	*11.5*	**10.2**	12.2	**0.695**	*0.695*	0.669
coke	**15.5**	23.7	*23.6*	**0.562**	*0.474*	0.452
bolt	*368.0*	**173.9**	390.7	*0.019*	**0.405**	0.017
boy	3.9	**3.2**	*3.3*	0.732	*0.773*	**0.775**
crossing	2.8	**1.9**	*2.8*	0.713	**0.789**	*0.742*
couple	12.4	**9.3**	*11.0*	*0.525*	**0.552**	0.474
football1	*6.3*	**5.7**	6.8	*0.624*	**0.653**	0.607
jogging-1	20.3	*19.1*	**18.4**	0.523	*0.555*	**0.577**
jogging-2	*8.8*	**6.2**	66.2	*0.750*	**0.759**	0.113
doll	*5.6*	**5.0**	5.6	*0.603*	**0.609**	0.548
girl	*6.4*	**5.1**	8.3	0.587	**0.654**	*0.599*
walking2	*3.9*	**3.1**	5.0	*0.578*	**0.632**	0.492
walking	*3.9*	**2.8**	7.3	*0.591*	**0.648**	0.566
david3	56.2	*11.2*	**10.1**	0.560	*0.718*	**0.734**
carScale	**11.5**	15.2	*14.6*	**0.515**	*0.497*	0.395
skiing	102.6	*86.0*	**5.4**	0.317	*0.327*	**0.493**
motorRolling	*61.6*	**36.6**	143.7	*0.458*	**0.527**	0.165
mountainBike	**5.9**	*6.0*	10.8	*0.762*	**0.773**	0.640
lemming	*11.0*	**10.6**	14.0	*0.691*	**0.701**	0.654
liquor	31.0	**24.2**	*27.4*	0.615	**0.704**	*0.696*
woman	**5.2**	*5.5*	12.7	**0.747**	*0.746*	0.703
faceocc1	57.1	*56.8*	**52.1**	*0.443*	0.440	**0.474**
basketball	*22.5*	**11.7**	23.0	0.522	**0.621**	*0.548*
subway	*5.2*	**5.0**	5.8	*0.735*	**0.739**	0.723
tiger1	57.2	*56.9*	**22.5**	0.215	*0.215*	**0.531**
tiger2	*15.1*	**14.9**	15.8	*0.636*	**0.638**	0.626
car4	28.1	**27.4**	*27.9*	**0.471**	*0.467*	0.407
sylvester	*5.7*	7.4	**5.3**	*0.732*	0.724	**0.742**
suv	**11.8**	14.1	*12.1*	**0.735**	0.654	*0.729*
jumping	6.6	**4.3**	*5.7*	0.648	**0.738**	*0.649*
fleetface	20.3	*19.5*	**16.4**	*0.608*	0.595	**0.657**
freeman1	**6.7**	*8.4*	11.5	**0.620**	*0.544*	0.390
freeman3	**6.4**	*10.7*	36.4	**0.436**	*0.429*	0.211
dog1	**4.0**	*4.2*	8.3	**0.659**	*0.648*	0.546
freeman4	*34.9*	59.0	**24.0**	**0.299**	0.162	*0.226*
football	*9.0*	**8.9**	20.5	*0.653*	**0.665**	0.579
faceocc2	*10.4*	**9.8**	10.6	*0.714*	**0.722**	0.706
fish	4.0	*3.7*	**3.1**	0.855	**0.871**	*0.866*
dudek	*10.9*	11.0	**10.4**	0.721	*0.722*	**0.734**
david2	*1.6*	**1.5**	1.8	0.818	**0.857**	*0.848*
mhyang	3.6	**2.5**	*3.2*	0.768	*0.806*	**0.808**
Mean	*24.0*	**18.3**	26.8	*0.595*	**0.619**	0.551
#Best score	11	28	12	12	28	11
#Second best score	23	15	13	24	17	10

Table 10: Center error (second through fourth columns) and overlap ratio (right three columns) of the proposed automatic and manual integration of segmentation with MIST for individual sequences. #Best and #Second best scores are the total number sequences that each of the 3 trackers (MIST, MIST-SEGA, and MIST-SEGM) performs best and second best, respectively.

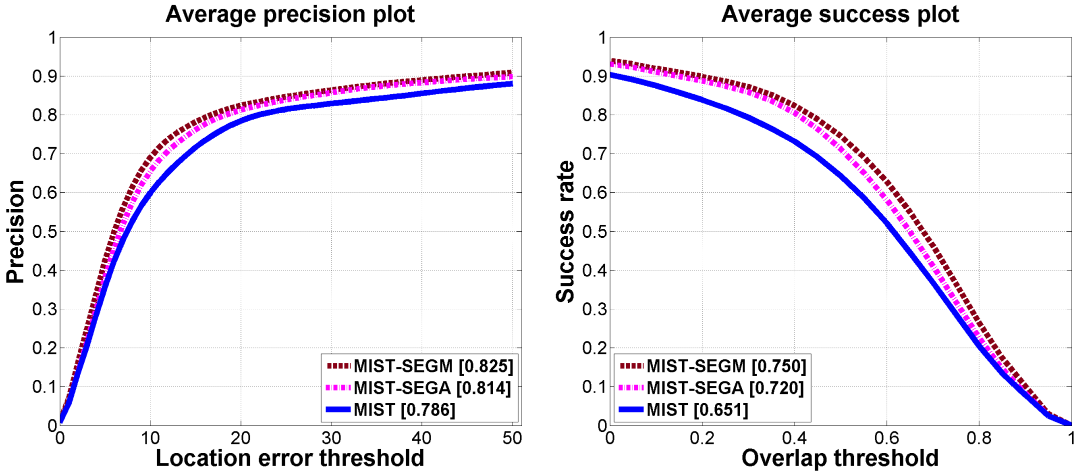

Figure 25: The averaged precision and success plots of the proposed automatic and manual integration of segmentation with MIST on all 50 sequences.

Tracker	Mean overlap score	Mean center error	Mean number of failures
MIST-SEGA	*0.481*	*31.08*	3.6

Table 11: Objective validation of tracking failure detection using the 10 test video sequences.

4.4.4.1 Object deformation, occlusion, and motion blur

We use *jogging-2*, *motorRolling*, and *skating1* video sequences to compare the proposed MIST-SEGM and MIST-SEGA with the proposed MIST during occlusion, deformation, and motion blur. As evidenced in the *jogging-2*, the inaccuracy cases of the proposed MIST due to implicit occlusion detection can be overcome by the proposed MIST-SEGM and MIST-SEGA. In the *motorRolling* sequence, the proposed MIST-SEGM and MIST-SEGA are more effective in handling motion blur and deformation.

4.4.4.2 Object scale variation, out-of-view, and fast motion

Figure 29 depicts qualitative results for *carScale*, *ironman*, and *singer1* sequences with scale variation, out-of-view, and fast motion. In the *carScale* and *singer1* sequences, we observe that the proposed MIST-SEGM and MIST-SEGA are more effective tracking objects undergoing scale variations.

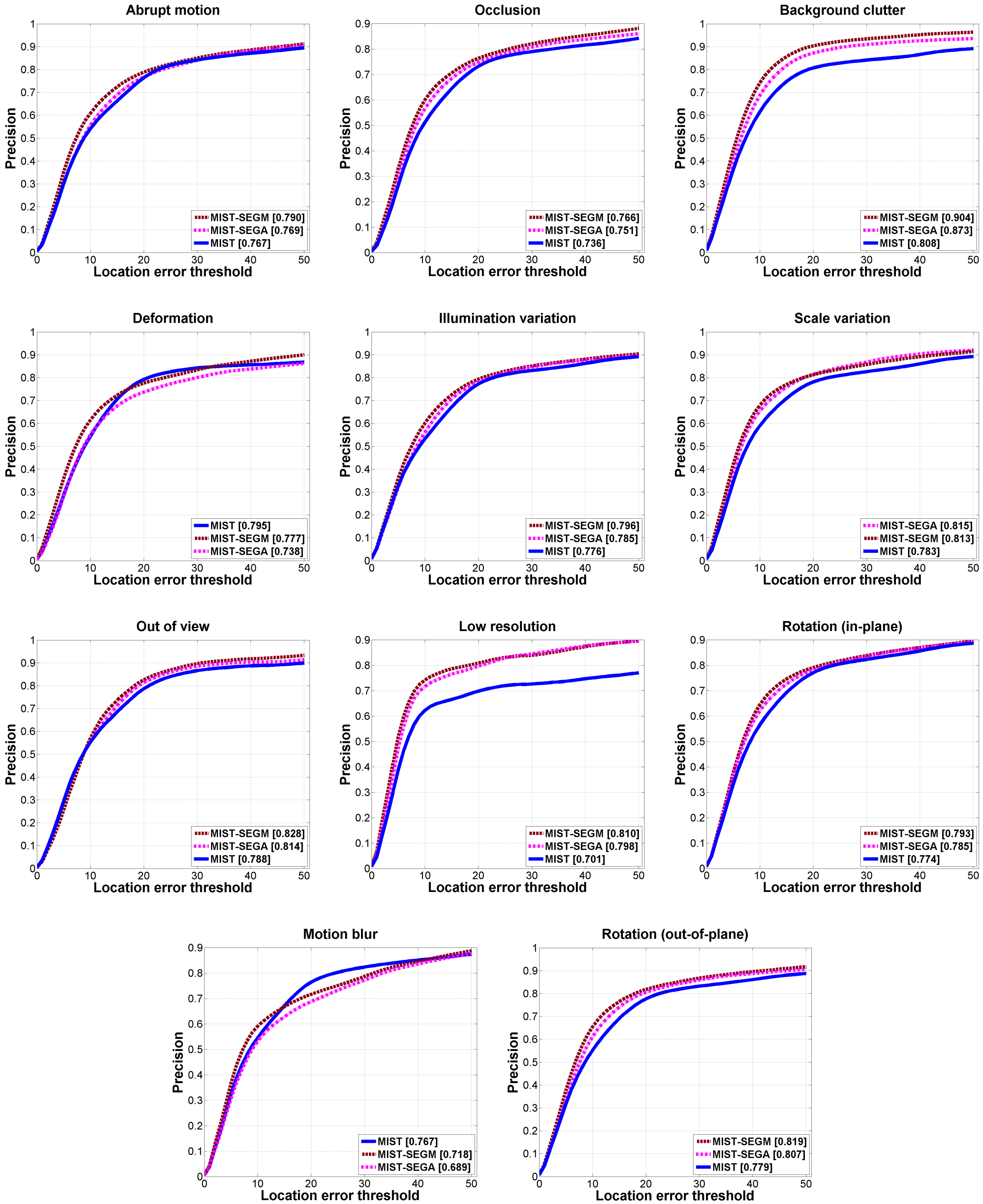

Figure 26: Precision plots for 11 challenging categories on all 50 sequences for comparing the proposed automatic and manual integration of segmentation with MIST.

4.4.4.3 Low resolution, illumination variation, and background clutter

In Figure 30, we subjectively compare the proposed two tracking methods using *walking*, *shaking*, and *coke* sequences in challenging scenarios, such as low resolution, illumination

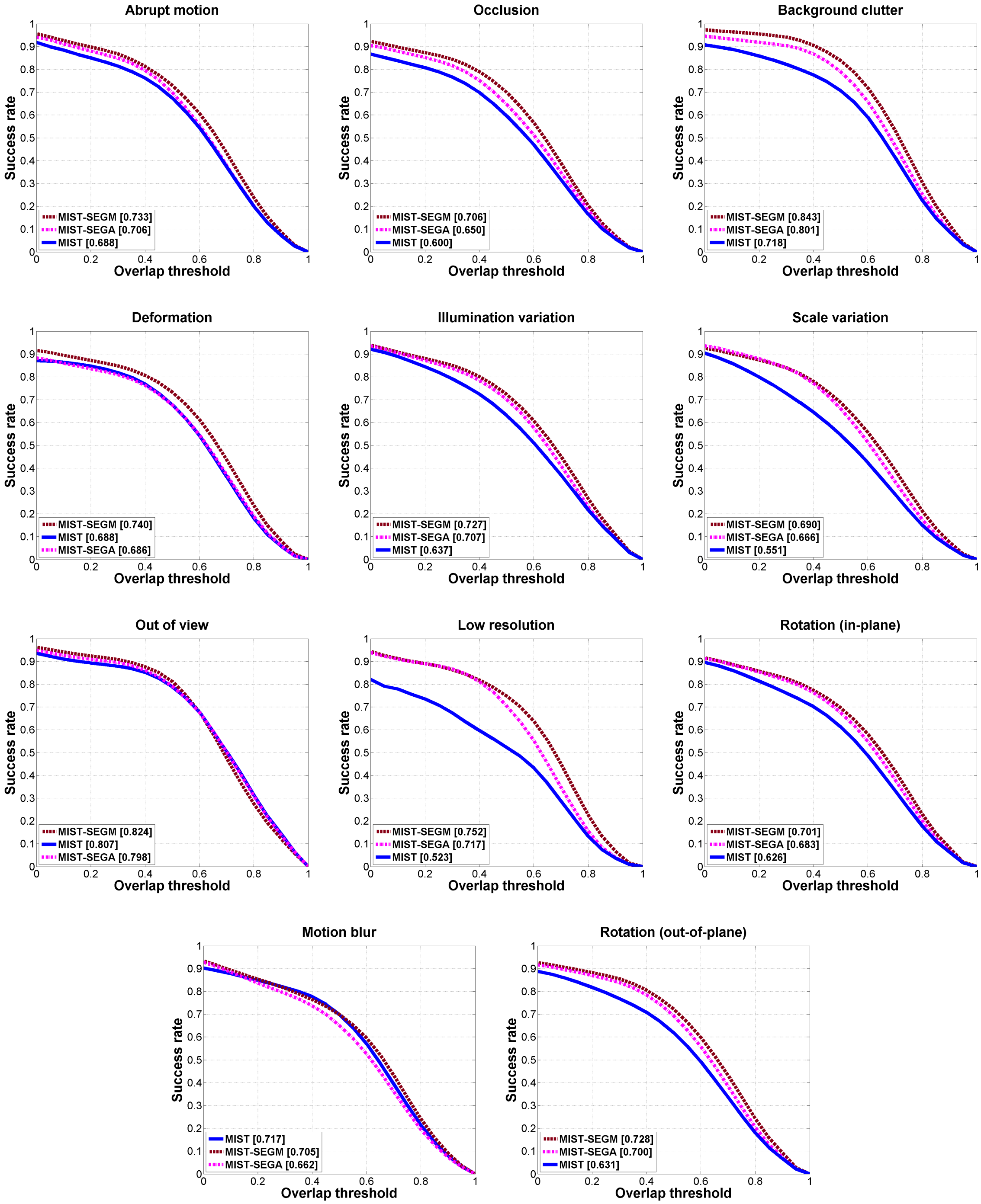

Figure 27: Success plots for 11 challenging categories on all 50 sequences for comparing the proposed automatic and manual integration of segmentation with MIST.

variation, background clutter, etc. In particular, tracking the object in *shaking* sequence is distracted by drastic illumination variation as well as object deformation. However, with the aid of the proposed tracking failure detection and tracking re-initialization, the proposed MIST-SEGM and MIST-SEGA are more effective tracking the target.

4.4.4.4 In-plane and out-plane rotation

The overall effectiveness of the proposed MIST-SEGM and MIST-SEGA in handling with object in-plane and out-plane rotation is subjectively demonstrated in Figure 31. In the *freeman3* sequence, the object is challenging to track due to relatively smaller BB initialized at the first frame. An appearance model sufficient to handle drastic scale variation in addition to the object rotation is difficult to be modeled from such a smaller BB. The effectiveness of the proposed integration of segmentation is clearly evident by this challenging sequence.

4.4.4.5 Segmentation output

Figure 32 shows some subjective results of MIST-SEGA with few sequences. As can be seen, BB extracted from the binary mask of the segmentation output is subjectively reliable for re-initialization of the proposed tracker.

4.4.4.6 Challenging scenarios

Figure 33 shows some challenging scenarios of the proposed MIST-SEGM and MIST-SEGA. In the *skiing* and *faceocc1* sequences, the proposed detection of tracking failure is suboptimal with gradual illumination changes, occlusion, and background clutter. This is because our conditional model update scheme incorporates part of the background. Incorporating an explicit occlusion detection technique can improve the proposed MIST-SEGM and MIST-SEGA, specially in the challenging situations similar to those encountered in *skiing*, *football*, and *faceocc1* sequences. However, we note that the proposed integration MIST and its integration with object segmentation are stable signifying that they do not significantly fail under any of the 50 videos used.

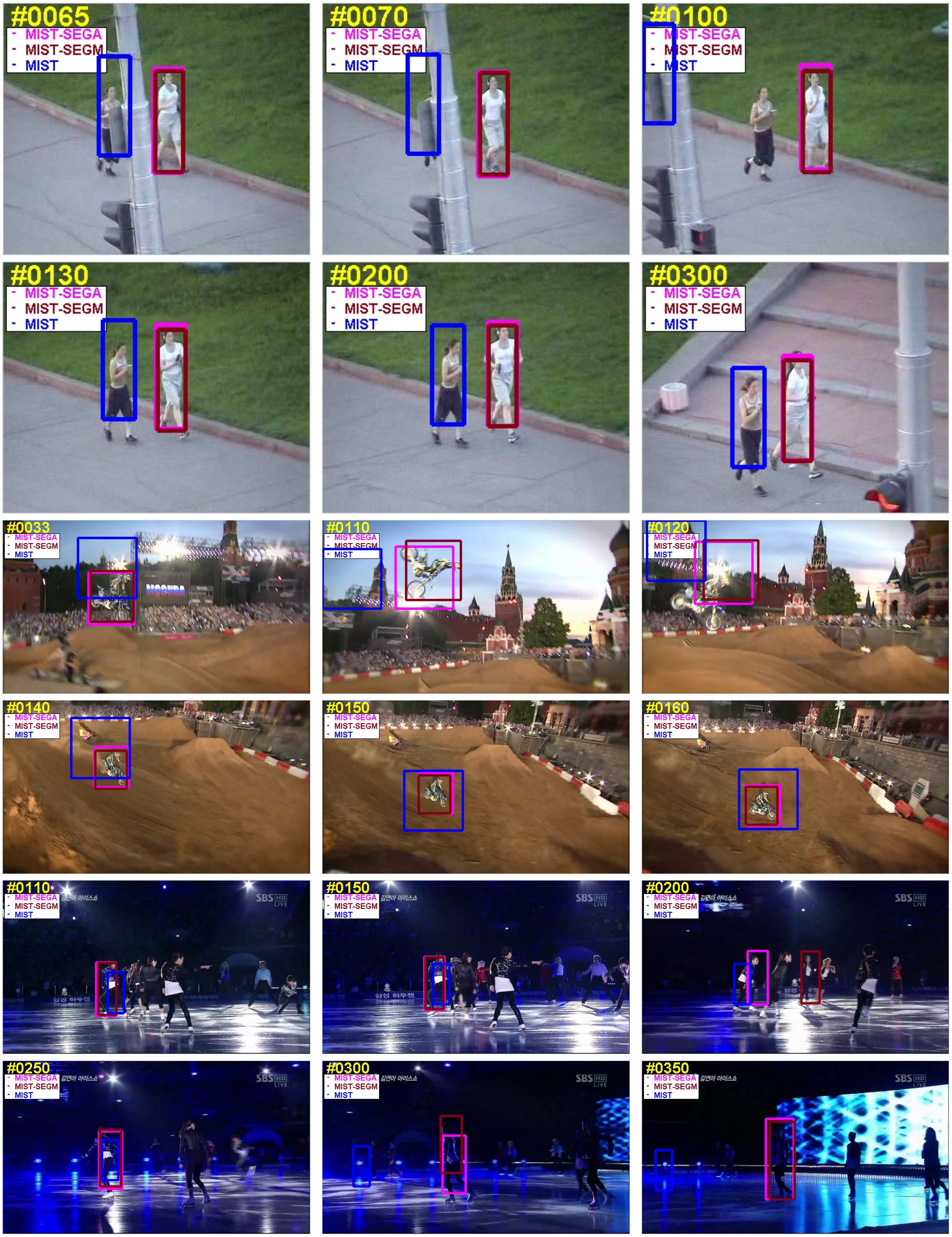

Figure 28: Improved tracking under occlusion, deformation, and motion blur. Tow *jogging-2*; middle *motorRolling*; and bottom *skating1*.

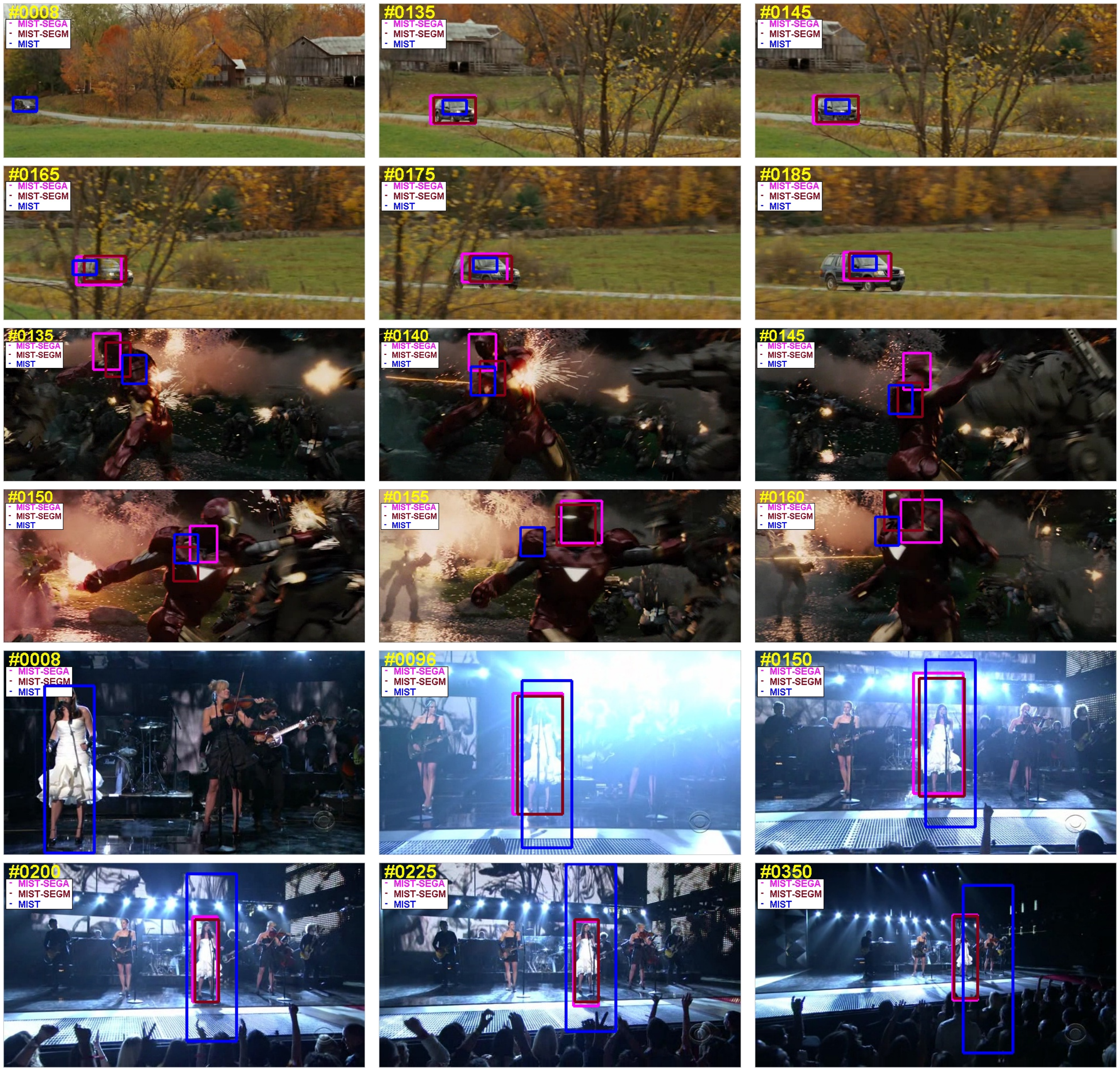

Figure 29: The proposed segmentation integration improves tracking under scale variation, out-of-view, and fast motion. Top *carScale*; middle *ironman*; and bottom *singer1*.

4.5 Conclusion

The simulation results in this Chapter show that the proposed object tracking method strongly competes against state-of-the-art trackers due to *a*) the proposed dynamic modeling by harmonic means and particle filter with entropy-based observation likelihood distribution, which effectively improves object localization, *b*) the proposed probability formulation on determining model updates for structured maximization problem, *c*) our adaptive weighted joint kernel function, which allows generalizing the tracking problem more effectively, and *d*) the

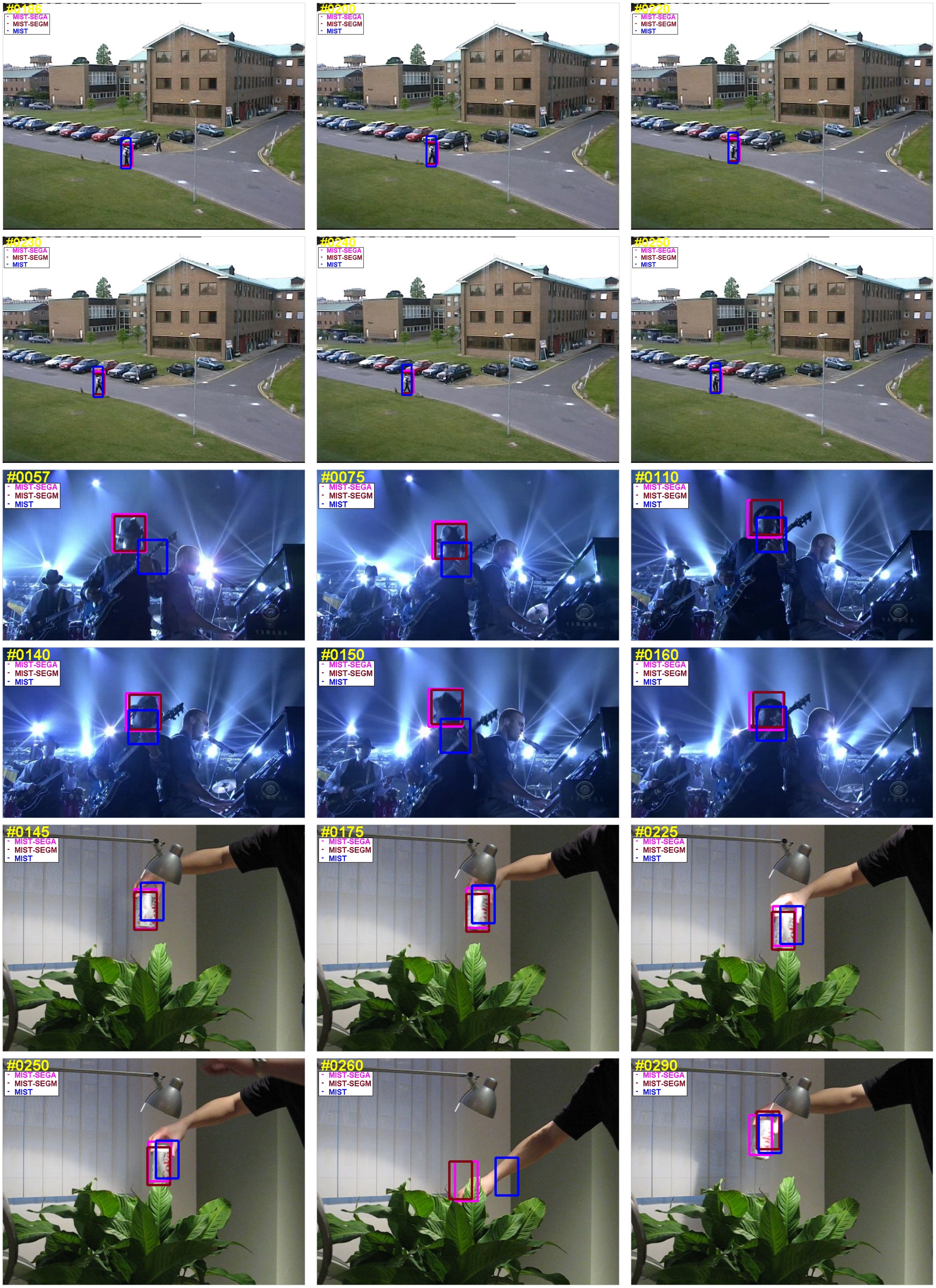

Figure 30: Improved tracking results with segmentation integration: *walking*, *shaking*, and *coke* with low resolution, illumination variation, and background clutter.

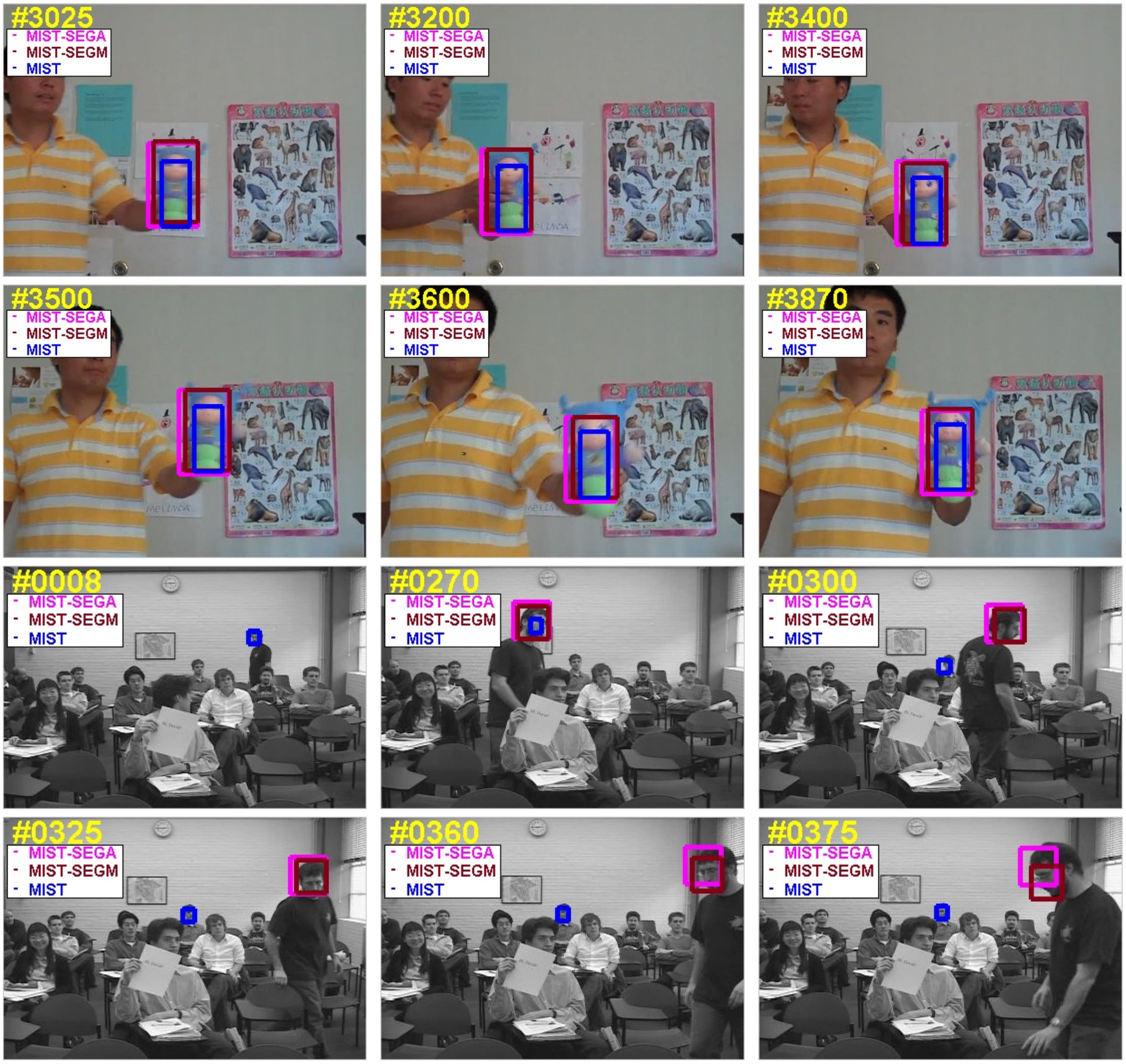

Figure 31: Tracking under in-plane and out-plane rotation is improved with segmentation integration. Top *doll* and bottom *freeman3*.

proposed motion-augmented regularization term, which constrains the output search space during inference.

Further improvements of the proposed object tracking method were attained by our tracking failure detection technique and the proposed integration of an active contour based segmentation method using particle filter to reinitialize the tracker. As expected, the choice of the segmentation method strongly affects performance of its integration in object tracking. Comparing supervised (manual) and unsupervised (automatic), BB selection has major impact as well. Both of these observations are complementary and future advancement in segmentation

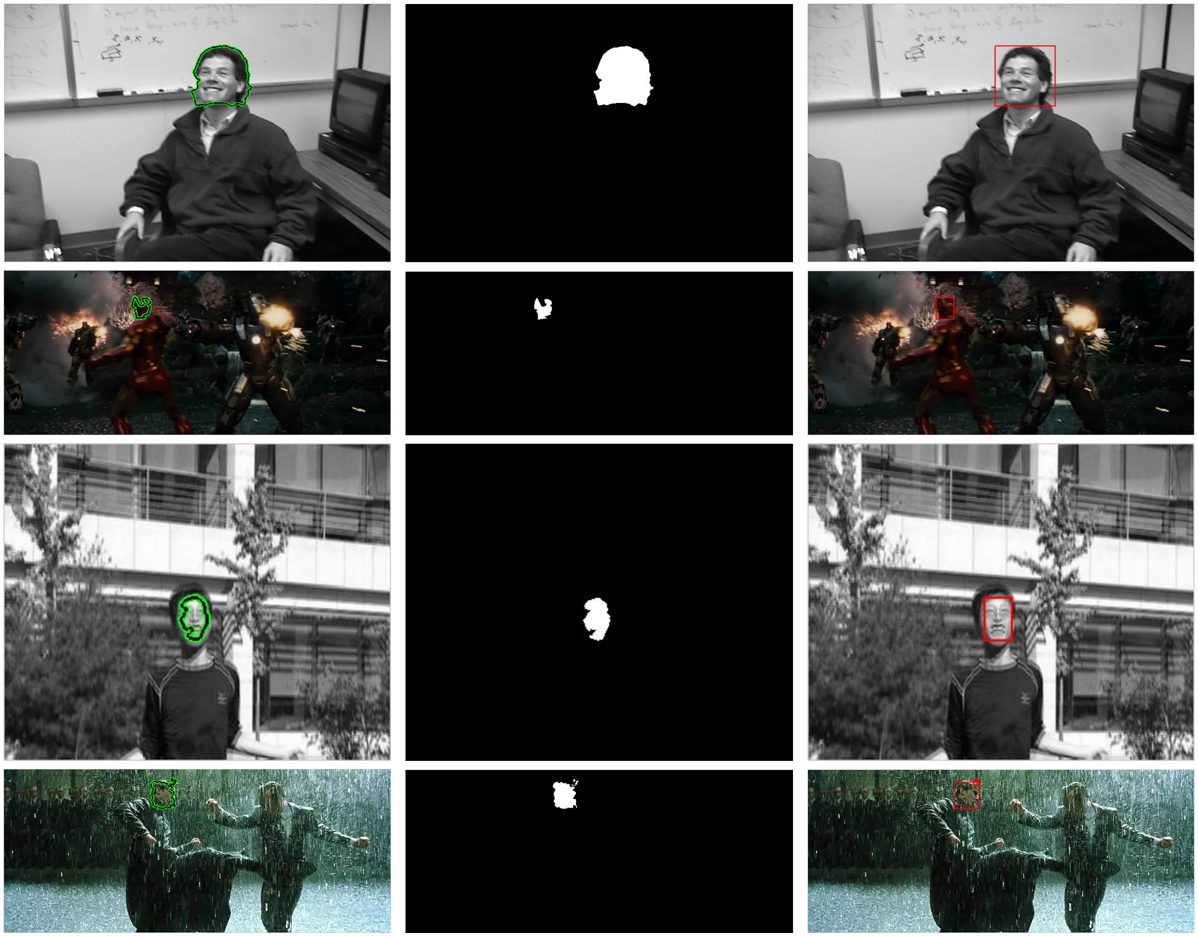

Figure 32: Segmentation outputs of *fleetface*, *ironman*, *jumping*, and *matrix* sequences. Left column – contours; middle column – binary masks; last column – BBs extracted from the binary mask.

will enhance object tracking. We, however, think that using current segmentation methods, such as active contour based [100], with the proposed automated integration (failure detection and BB selection) significantly improve object tracking (for example, MIST becomes better in 36 versus 21 videos, cf. Table 10).

In Table 12, we summarize main features of our techniques and several state-of-the-art trackers. Some of the highlights includes *a*) our methods incorporate color features through the proposed JKF formulation, which is largely ignored by the compared methods, *b*) for sampling, only our methods, *ASLA*, and *SCM* employ particle filter while other methods use inefficient dense sampling strategies, and *c*) only the proposed MIST-SEG and *TLD* employ tracking failure and recovery. The proposed MIST-SEG implicitly handles scale variations

Tracker	Features	Search	Dynamic model	Scale	FDR
Struck [66]	Haar	Dense search	×	×	×
TLD [65]	Points	Dense search	Median flow	✓	✓
ASLA [121]	Sparse codes	Particle filter	Autoregressive	✓	×
CSK [122]	Template	Dense search	×	✓	×
SCM [87]	Sparse codes	Particle filter	Affine	✓	×
MIST [Ours]	HoG + Color	Particle filter	KHM	×	×
MIST-SEG [Ours]	HoG + Color	Particle filter	KHM	✓⃝	✓

Table 12: Comparison of our methods with several state-of-the-art trackers. Under Scale column, the symbol ✓⃝ indicates that our MIST-SEG tracker partially supports scale variation with segmentation integration. FDR denotes failure tracking detection and recovery.

when re-initializing tracking using the proposed technique of object segmentation (see Section 3.3.3.1), where we estimate both the location and size of the object once failure is detected.

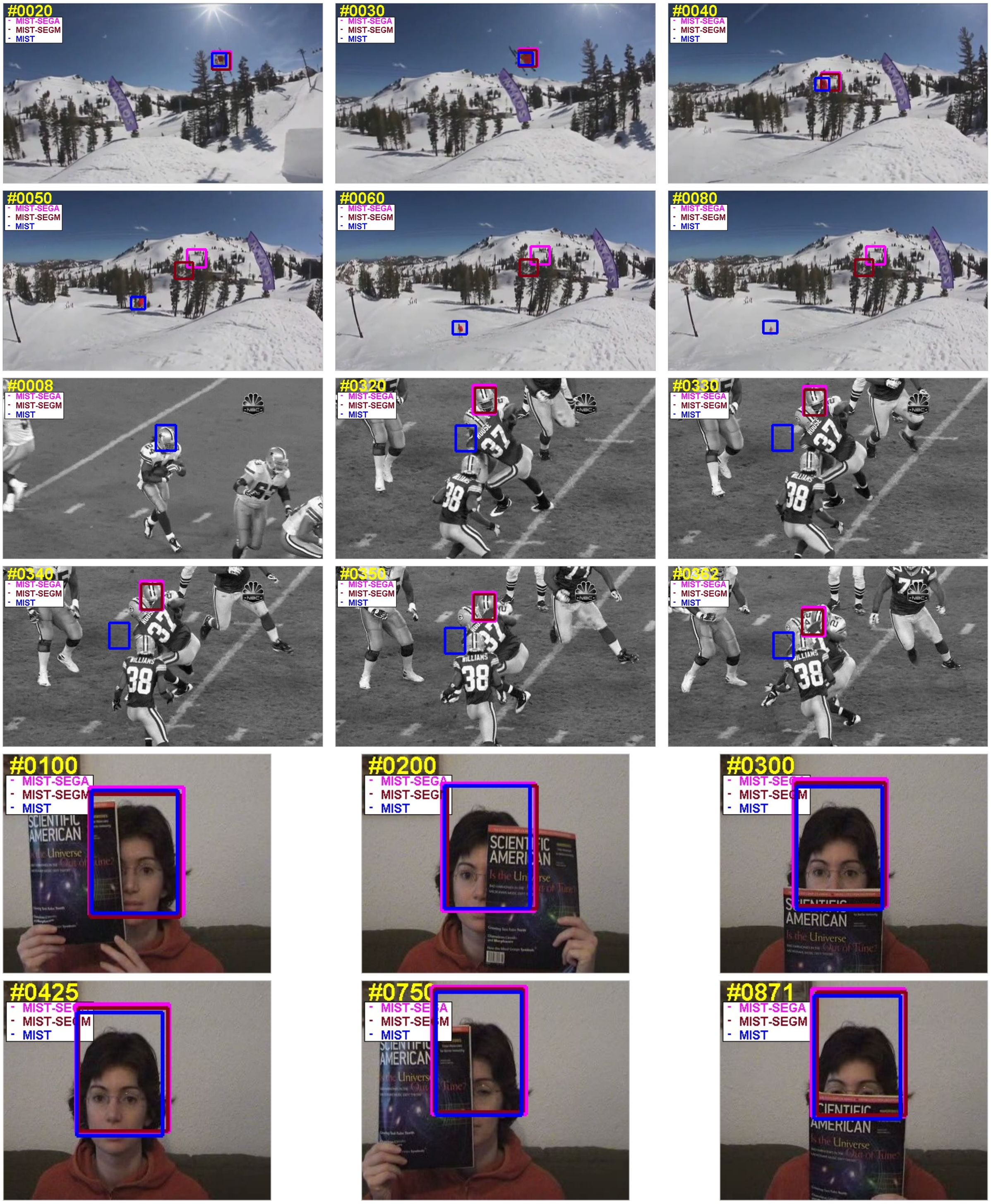

Figure 33: Some challenging scenarios of tracking with integrated segmentation: *skiing*, *football*, and *faceocc1* with occlusion, drastic illumination variation, and rotation.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Object tracking has gained increased attention in both academia and industry due to its widespread applications including augmented and virtual reality, human-computer interaction, automated navigation systems as drones or self-driving cars, and social video content analysis. However, object tracking is a difficult problem inevitably causing frequent tracking failures due to many challenges inherited in video sequences, such as deformation, illumination changes, occlusion, and background clutter. Our related work study showed that traditional trackers without explicit appearance modeling are suboptimal under challenging conditions. As such, online appearance modeling based on machine learning theory has attracted significant attention to account for intrinsic appearance changes. Based on various appearance modeling, online machine learning based object tracking can be categorized into three classes: generative, discriminative, and hybrid generative-discriminative methods. Discriminative object tracking methods are more effective, compared with other object tracking categories based on machine learning. This is because discriminative methods compute a decision boundary that can optimally separate the object from the background. Consequently, much of current object tracking research focuses on discriminative category.

Our contributions in this thesis are two object tracking methods. We built our baseline method using stochastic processes and machine learning theory. We extended this baseline method by effectively integrating object segmentation. Specifically, we contributed

1. a method to represent the target dynamics as a random stochastic process using harmonic means and particle filter for predicting it;
2. a formulation to model a new observation likelihood model for the particle filter by using kernel machines and entropy to evaluate certainty of the likelihood distribution;
3. an adaptive weighted joint kernel function to construct an effective appearance model;
4. a probability formulation to determine model updates for structured maximization problem;
5. a motion-augmented regularization term during inference to constrain the output search space;
6. a technique to detect tracking failures based on online binary support vector machines framework; and
7. a particle filter based automated method to re-initialize the tracker based on an active contour based object segmentation.

Our baseline object tracking method is effective under several challenges by employing advanced techniques in machine learning: an adaptive dynamic model and a structured support vector machines framework. In the proposed method, *first*, we modeled the target dynamics as a random stochastic process, and adopted harmonic means and particle filter to predict dynamics. In our dynamic model, we introduced a new observation likelihood model using kernel machines. We used entropy to evaluate certainty of our observation likelihood distribution. *Second*, we used online structured support vector machines to the tracking problem because they can be generalized well through the use of kernels, while being effective against estimation noise. For modeling the target appearance, we developed an adaptive weighted joint kernel function using color and histogram of gradients as feature descriptors. For learning, we built a probability model to avoid model updates when the target is absent from the scene. To gain computational efficiency improvements, we used particle filter for sampling instead of exhaustive dense sampling, and introduced a motion-augmented regularization term during inference to constrain the output search space. Through extensive experiments, we demonstrated that the proposed computationally efficient object tracking method well competes with state-of-the-art trackers on standard datasets, and our technique is more effective against many challenges often encountered in real-world applications.

Tracking failures or inaccuracies are inevitable; therefore, effective tracking requires both detecting tracking failures (or inaccuracies) and re-initialization after failures. To that end, we extended our baseline tracker and proposed a method that integrates object segmentation into tracking to minimize tracking failures thereby improving overall accuracy in object tracking. In our integrated method, *first*, we proposed a technique to detect tracking failures based on online binary support vector machines framework. *Second*, to recover from tracking failures, we proposed an automated method to re-initialize the tracker based on an active contour based object segmentation. We used particle filter to automatically select bounding box for segmentation. Through experiments, we observe that the choice of the segmentation method strongly affects performance of its integration in object tracking, and a different object tracking method may differently benefit from the integration of the same segmentation method. Comparing supervised and unsupervised, bounding box selection has major impact as well. These observations are complementary and future advancement in segmentation will enhance object tracking. Our experiments showed that using state-of-the-art segmentation methods (without adaptation) with the proposed automated integration of failure detection and bounding box selection well improve our baseline object tracking method.

5.2 Future Work

5.2.1 Online Object Tracking

When characterizing the object of interest and the relevant background (i.e., positive and negative samples, respectively) for the structured support vector machines framework, the proposed baseline method gives equal or more importance to the negative samples. In order to keep computational demand and storage low, we collect only a few negative samples each frame from different locations although there is virtually unlimited amount of negative samples available throughout the sequence. These limited samples can inhibit the effectiveness of long term tracking. Thus, future work could include exploring how to leverage these vast amount of negative samples, for example, by using correlation filters which offer the ability to simultaneously localize and classify the object of interest [75, 131, 132].

Our method employs color and histogram of gradients as feature descriptors and predefined kernels within the structured learning framework. A potential avenue for future work would be to investigate using additional features, for example, Haar-like features [71], ORB [133], BRISK [134], FREAK [135], BRIEF [136], etc. To that end, a multiple kernel learning [130] framework can be employed to adaptively combine both kernels and their parameters optimally. To tackle the high dimensional feature space, often encountered when using multiple kernel learning for tracking, a dimensionality reduction approach, such as principal component analysis [137] can also be investigated.

Depth data at each pixel level, in addition to color, is available from most recent commodity RGB-D cameras [2,4]. One prospective avenue for future work would be to incorporate the depth data into the proposed tracking method by extending our joint kernel formulation. Such work can benefit many applications specially in augmented and virtual reality, for example, entertainment, education, and manufacturing [4].

Another interesting extension of the proposed method would be to handle scale variations of deformable objects, which could substantially improve the accuracy of our tracker. One avenue for future work could be to model and learn features of deformable and articulated objects or model the deformable object using many smaller objects.

From implementation perspectives, a potential future work would be to implement the proposed method on a hardware platform, such as on FPGAs, which can significantly improve the efficiency. Such hardware implementation can be combined with our proposed hardware architecture for object segmentation [29] for applications utilizing stationary cameras.

Our object tracking technique is currently implemented for tracking a single object. The proposed method can be extended to track multiple objects by engaging several single instances of the proposed trackers. Alternatively, a linear programming model, such as [138], can be utilized to handle jointly tracking multiple objects by taking inter-object constraints and layout information into account.

5.2.2 Integration of Segmentation into Tracking

Applying object tracking for improving object segmentation is an active research field [107–110]. As such, an interesting area of future work would be to study how the proposed tracker can be employed to enhance the quality of video object segmentation.

A potential future work that could extend the proposed tracker would be to produce more accurate articulated object boundaries in-lieu of the current rectangular bounding box as the tracker's output. The proposed integration of segmentation with tracking facilitates developing such work. However, popular tracking benchmark methodologies, for example [7–9], are based on rectangular bounding box. To our best knowledge, a platform that evaluates trackers based on articulated object boundaries on large datasets has not been reported in literature. Thus, our proposed research implicitly motivates devising such evaluation benchmark along with new evaluation metrics.

We have investigated the integration of segmentation into our own tracker. An interesting future work could include testing similar, but adaptive, integration of segmentation into other trackers, such as *Struck* [66] or *ASLA* [121].

Object segmentation is usually error-prone, especially during object occlusion. A potential extension of the proposed tracker would be to measure when the object segmentation fails and consequently avoid relying on it for calculating the likelihood distribution. In such scenarios, BB can be used to extract the likelihood distribution instead of using segmentation output.

Bibliography

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[2] K. Konolige, “Projected texture stereo,” in *Robotics and Automation, IEEE International Conference on*, 2010, pp. 148–155.

[3] 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, 2011.

[4] H.-L. Chi, S.-C. Kang, and X. Wang, “Research trends and opportunities of augmented reality applications in architecture, engineering, and construction,” *Automation in construction*, vol. 33, pp. 116–122, 2013.

[5] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti, “Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking,” *Signal Processing Magazine, IEEE*, vol. 22, no. 2, pp. 38–51, 2005.

[6] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: an experimental survey,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 7, pp. 1442–1468, 2014.

[7] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2013, pp. 2411–2418.

[8] Y. Wu, J. Lim, and M.-H. Yang, “Object tracking benchmark,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 9, pp. 1834–1848, 2015.

[9] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, “The visual object tracking VOT2015 challenge results,” in *Computer Vision Workshops, IEEE International Conference on*, 2015, pp. 1–23.

[10] A. Li, M. Lin, Y. Wu, M. Yang, and S. Yan, “Nus-pro: A new visual tracking challenge.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 38, no. 2, p. 335, 2016.

[11] S. Weng, C. Kuo, and S. Tu, “Video object tracking using adaptive kalman filter,” *Journal of Visual Communication and Image Representation*, vol. 17, no. 6, pp. 1190–1208, 2006.

[12] M. Mirabi and S. Javadi, “People tracking in outdoor environment using kalman filter,” *Intelligent Systems, Modelling and Simulation, IEEE International Conference on*, pp. 303–307, 2012.

[13] Y. Rui and Y. Chen, “Better proposal distributions: Object tracking using unscented particle filter,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. 786–793, 2001.

[14] M. Isard and A. Blake, “Condensation - conditional density propagation for visual tracking,” *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.

[15] A. Doucet, N. De Freitas, N. Gordon *et al.*, *Sequential Monte Carlo methods in practice*. Springer New York, 2001.

[16] K. Nummiaro, E. Koller-Meier, and L. Van Gool, “An adaptive color-based particle filter,” *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.

[17] G. Bradski, “Computer vision face tracking for use in a perceptual user interface,” *Intel Technology Journal*, vol. 2, no. 2, pp. 12–21, 1998.

[18] I. Haritaoglu, D. Harwood, and L. Davis, “W4: Real-time surveillance of people and their activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 809–830, 2000.

[19] J. Kang, I. Cohen, and G. Medioni, "Continuous tracking within and across camera streams," *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, pp. 267–272, 2003.

[20] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 11, pp. 1531–1536, 2004.

[21] C. Shan, Y. Wei, T. Tan, and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift," *Automatic Face and Gesture Recognition, IEEE International Conference on*, pp. 669–674, 2004.

[22] K. Shafique and M. Shah, "A noniterative greedy algorithm for multiframe point correspondence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 1, pp. 51–65, 2005.

[23] Y. Hou, H. Sahli, R. Ilse, Y. Zhang, and R. Zhao, "Robust shape-based head tracking," *Advanced Concepts for Intelligent Vision Systems*, pp. 340–351, 2007.

[24] Z. Wang, X. Yang, Y. Xu, and S. Yu, "Camshift guided particle filter for visual tracking," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 407–413, 2009.

[25] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[26] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *Image Processing, IEEE Transactions on*, vol. 13, no. 11, pp. 1491–1506, 2004.

[27] M. Yin, J. Zhang, H. Sun, and W. Gu, "Multi-cue-based CamShift guided particle filter tracking," *Expert Systems with Applications*, vol. 38, no. 5, pp. 6313–6318, 2011.

[28] E. Erdem, S. Dubuisson, and I. Bloch, "Visual tracking by fusing multiple cues with context-sensitive reliabilities," *Pattern Recognition*, vol. 45, no. 5, pp. 1948–1959, 2012.

[29] K. Ratnayake and A. Amer, “Embedded architecture for noise-adaptive video object detection using parameter-compressed background modeling,” *Journal of Real-Time Image Processing*, pp. 1–18.

[30] K. Ratnayake and A. Amer, “Object Tracking with Adaptive Motion Modeling of Particle Filter and Support Vector Machines,” in *Image Processing, IEEE International Conference on*, 2015, pp. 1140–1144.

[31] I. Sethi and R. Jain, “Finding trajectories of feature points in a monocular image sequence,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 1, pp. 56–73, 1987.

[32] K. Rangarajan and M. Shah, “Establishing motion correspondence,” *CVGIP: image understanding*, vol. 54, no. 1, pp. 56–73, 1991.

[33] C. Veenman, M. Reinders, and E. Backer, “Resolving motion correspondence for densely moving points,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 1, pp. 54–72, 2001.

[34] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–575, 2003.

[35] H. Schweitzer, J. Bell, and F. Wu, “Very fast template matching,” *Computer Vision*, pp. 145–148, 2006.

[36] D. Rowe, I. Reid, J. Gonzàlez, and J. Villanueva, “Unconstrained multiple-people tracking,” *Pattern Recognition*, pp. 505–514, 2006.

[37] H. Wang, L. Huo, and J. Zhang, “Target tracking algorithm based on dynamic template and kalman filter,” *Communication Software and Networks, IEEE Conference on*, pp. 330–333, 2011.

[38] A. Bhattacharyya, “On a measure of divergence between two multinomial populations,” *Sankhya: The Indian Journal of Statistics*, vol. 7, no. 4, pp. 401–406, 1946.

[39] T. Kailath, “The divergence and bhattacharyya distance measures in signal selection,” *Communication Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 52–60, 1967.

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

[41] R. Collins, “Mean-shift blob tracking through scale space,” *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 2, pp. 234–240, 2003.

[42] L. Bretzner and T. Lindeberg, “Feature tracking with automatic selection of spatial scales,” *Computer Vision and Image Understanding*, vol. 71, no. 3, pp. 385–392, 1998.

[43] Z. Zivkovic and B. Krose, “An EM-like algorithm for color-histogram-based object tracking,” *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, pp. 798–803, 2004.

[44] N. JIFENG, L. Zhang, D. Zhang, and C. Wu, “Robust object tracking using joint color-texture histogram,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 07, pp. 1245–1263, 2009.

[45] Z. Khan, T. Balch, and F. Dellaert, “A Rao-Blackwellized particle filter for Eigen-Tracking,” *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 2, pp. 980–986, 2004.

[46] W. Lu, K. Okuma, and J. Little, “Tracking and recognizing actions of multiple hockey players using the boosted particle filter,” *Image and Vision Computing*, vol. 27, no. 1, pp. 189–205, 2009.

[47] G. Casella and C. Robert, “Rao-Blackwellisation of sampling schemes,” *Biometrika*, vol. 83, no. 1, pp. 81–94, 1996.

[48] X. Linzhou, Z. Xin-hua, Y. Shao-qing, and F. Wen-tao, “An efficient particle filter with variable number of particles for bearings-only tracking,” *Signal Processing, IEEE International Conference on*, pp. 2395–2398, 2010.

[49] W. Hassan, N. Bangalore, P. Birch, R. Young, and C. Chatwin, “An adaptive sample count particle filter,” *Computer Vision and Image Understanding*, 2012.

[50] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

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

[52] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large margin methods for structured and interdependent output variables,” in *Journal of Machine Learning Research*, 2005, pp. 1453–1484.

[53] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[54] X. Mei and H. Ling, “Robust visual tracking and vehicle classification via sparse representation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 11, pp. 2259–2272, 2011.

[55] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, “Minimum error bounded efficient 1 tracker with occlusion detection,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2011, pp. 1257–1264.

[56] C. Bao, Y. Wu, H. Ling, and H. Ji, “Real time robust l_1 tracker using accelerated proximal gradient approach,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2012, pp. 1830–1837.

[57] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, “Robust tracking using local sparse appearance model and k-selection,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2011, pp. 1313–1320.

[58] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *European Conference on Computer Vision*, 2010, pp. 624–637.

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

[60] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[61] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[62] C. Tian, X. Gao, W. Wei, and H. Zheng, "Visual tracking based on the adaptive color attention tuned sparse generative object model," *Image Processing, IEEE Transactions on*, vol. 24, no. 12, pp. 5236–5248, 2015.

[63] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[64] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2009, pp. 983–990.

[65] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[66] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. Hicks, and P. Torr, "Struck: Structured output tracking with kernels," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[67] H. Grabner and H. Bischof, "On-line boosting and vision," in *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, 2006, pp. 260–267.

[68] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *European Conference on Computer Vision*, 2008, pp. 234–247.

[69] B. Babenko, M.-H. Yang, and S. Belongie, “Robust object tracking with online multiple instance learning,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[70] K. Zhang and H. Song, “Real-time visual tracking via online weighted multiple instance learning,” *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.

[71] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, 2001, pp. 511–518.

[72] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision.” in *Artificial Intelligence, International Joint Conference on*, 1981, pp. 674–679.

[73] J. Shi and C. Tomasi, “Good features to track,” in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[74] A. Bosch, A. Zisserman, and X. Munoz, “Image classification using random forests and ferns,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[75] R. C. González and R. E. Woods, *Digital image processing*. Prentice Hal, 2008.

[76] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, “Long-term correlation tracking,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015, pp. 5388–5396.

[77] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 3, pp. 583–596, 2015.

[78] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, “Learning spatially regularized correlation filters for visual tracking,” in *Computer Vision, IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318.

[79] M. Tang and J. Feng, “Multi-kernel correlation filter for visual tracking,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2015, pp. 3038–3046.

[80] W. Hu, J. Gao, J. Xing, C. Zhang, and S. Maybank, “Semi-supervised tensor-based graph embedding learning and its application to visual discriminant tracking,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2016.

[81] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, “Bing: Binarized normed gradients for objectness estimation at 300fps,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2014, pp. 3286–3293.

[82] P. Liang, C. Liao, X. Mei, and H. Ling, “Adaptive objectness for object tracking,” *IEEE Signal Processing Letters*, vol. PP, no. 99, pp. 1–1, 2016.

[83] A. Mazzu, P. Morerio, L. Marcenaro, and C. S. Regazzoni, “A cognitive control-inspired approach to object tracking,” *Image Processing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2016.

[84] A. Jordan, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes,” *Advances in neural information processing systems*, vol. 14, pp. 841–848, 2001.

[85] K. Zhang, L. Zhang, and M.-H. Yang, “Real-time compressive tracking,” in *European Conference on Computer Vision*. Springer, 2012, pp. 864–877.

[86] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, “Online discriminative object tracking with local sparse representation,” in *Applications of Computer Vision, IEEE Workshop on*, 2012, pp. 425–432.

[87] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparsity-based collaborative model,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2012, pp. 1838–1845.

[88] D. Fradkin and D. Madigan, “Experiments with random projections for machine learning,” in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 517–522.

[89] T. Bai, Y.-F. Li, and X. Zhou, “Learning local appearances with sparse representation for robust and fast visual tracking,” *Cybernetics, IEEE Transactions on*, vol. 45, no. 4, pp. 663–675, 2015.

[90] J. A. Lasserre, C. M. Bishop, and T. P. Minka, “Principled hybrids of generative and discriminative models,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, 2006, pp. 87–94.

[91] J. Mooser, S. You, and U. Neumann, “Real-time object tracking for augmented reality combining graph cuts and optical flow,” in *Mixed and Augmented Reality, IEEE and ACM International Symposium on*, 2007, pp. 1–8.

[92] J. Malcolm, Y. Rathi, and A. Tannenbaum, “Multi-object tracking through clutter using graph cuts,” in *Computer Vision, IEEE International Conference on*, 2007, pp. 1–5.

[93] N. Papadakis and A. Bugeau, “Tracking with occlusions via graph cuts,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 1, pp. 144–157, 2011.

[94] M. Godec, P. M. Roth, and H. Bischof, “Hough-based tracking of non-rigid objects,” *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1245–1256, 2013.

[95] T. F. Chan and L. A. Vese, “Active contours without edges,” *Image processing, IEEE transactions on*, vol. 10, no. 2, pp. 266–277, 2001.

[96] N. Paragios and R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 3, pp. 266–280, 2000.

[97] X. Zhou, X. Li, and W. Hu, “Level set tracking with dynamical shape priors,” in *Image Processing, IEEE International Conference on*, 2008, pp. 1540–1543.

[98] P. Chockalingam, N. Pradeep, and S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," in *Computer Vision, IEEE International Conference on*, 2009, pp. 1530–1537.

[99] J. Ning, L. Zhang, D. Zhang, and W. Yu, "Joint registration and active contour segmentation for object tracking," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 9, pp. 1589–1597, 2013.

[100] X.-F. Wang, D.-S. Huang, and H. Xu, "An efficient local Chan–Vese model for image segmentation," *Pattern Recognition*, vol. 43, no. 3, pp. 603–618, 2010.

[101] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," in *European Conference on Computer Vision*, 2008, pp. 264–275.

[102] K. E. Papoutsakis and A. A. Argyros, "Object tracking and segmentation in a closed loop," in *Advances in Visual Computing*. Springer, 2010, pp. 405–416.

[103] K. E. Papoutsakis and A. A. Argyros, "Integrating tracking with fine object segmentation," *Image and Vision Computing*, vol. 31, no. 10, pp. 771–785, 2013.

[104] J. Kwon and K. M. Lee, "Highly nonrigid object tracking via patch-based dynamic appearance modeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 10, pp. 2427–2441, 2013.

[105] Y. Huang, Y. Huang, and H. Niemann, "Segmentation-based object tracking using image warping and kalman filtering," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3, 2002, pp. 601–604.

[106] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 583–598, 1991.

[107] W. Brendel and S. Todorovic, "Video object segmentation by tracking regions," in *Computer Vision, IEEE International Conference on*, 2009, pp. 833–840.

[108] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” in *European Conference on Computer Vision*, 2010, pp. 282–295.

[109] P. Ochs and T. Brox, “Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions,” in *Computer Vision, IEEE International Conference on*, 2011, pp. 1583–1590.

[110] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin, “A novel performance evaluation methodology for single-target trackers,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[111] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[112] J. Platt, “Fast training of support vector machines using sequential minimal optimization,” *Advances in kernel methods-support vector learning*, vol. 3, 1999.

[113] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[114] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[115] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[116] A. Bordes, L. Bottou, P. Gallinari, and J. Weston, “Solving multiclass support vector machines with LaRank,” in *International Conference on Machine learning*, 2007, pp. 89–96.

[117] M. B. Blaschko and C. H. Lampert, “Learning to localize objects with structured output regression,” in *European Conference on Computer Vision*. Springer, 2008, pp. 2–15.

[118] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *Transactions on Graphics*, vol. 23, no. 3, pp. 303–308, 2004.

[119] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, 2002.

[120] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006.

[121] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2012, pp. 1822–1829.

[122] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European Conference on Computer Vision*. Springer, 2012, pp. 702–715.

[123] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, pp. 798–805, 2006.

[124] Q. Wang, F. Chen, J. Yang, W. Xu, and M.-H. Yang, "Transferring visual prior for online object tracking," *Image Processing, IEEE Transactions on*, vol. 21, no. 7, pp. 3296–3305, 2012.

[125] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2013, pp. 2371–2378.

[126] D. Wang, H. Lu, Z. Xiao, and Y.-W. Chen, "Fast and effective color-based object tracking by boosted color distribution," *Pattern Analysis and Applications*, vol. 16, no. 4, pp. 647–661, 2013.

[127] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, “Visual tracking via discriminative sparse similarity map,” *Image Processing, IEEE Transactions on*, vol. 23, no. 4, pp. 1872–1881, April 2014.

[128] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, “Adaptive color attributes for real-time visual tracking,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2014.

[129] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in *Computer Vision*. Springer, 2002, pp. 661–675.

[130] M. Gönen and E. Alpaydın, “Multiple kernel learning algorithms,” *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[131] D. S. Bolme, B. A. Draper, and J. R. Beveridge, “Average of synthetic exact filters,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2009, pp. 2105–2112.

[132] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2010, pp. 2544–2550.

[133] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *Computer Vision, IEEE International Conference on*, 2011, pp. 2564–2571.

[134] S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *Computer Vision, IEEE International Conference on*, 2011, pp. 2548–2555.

[135] A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2012, pp. 510–517.

[136] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *European Conference on Computer Vision*, 2010, pp. 778–792.

[137] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[138] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2007, pp. 1–8.