Pattern Extraction by Modeling Image Spatial Relationship

PATTERN EXTRACTION BY MODELING IMAGE SPATIAL RELATIONSHIP

BY

YUANHAO YU, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2017) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Pattern Extraction by Modeling Image Spatial Relationship
AUTHOR:	Yuanhao Yu M.Sc., (Pattern Recognition and Intelligent System) Institute of Automation, Chinese Academy of Sciences, Beijing, China
SUPERVISOR:	Dr. T. Kirubarajan
NUMBER OF PAGES:	xvi, 137

To my family

Abstract

In this thesis, a universal framework that is able to extract image spatial relationship among multiple appearance components is proposed, which can be employed to extract additional pattern in wide computer vision tasks. In order to demonstrate its usefulness, three novel algorithms solving different computer vision problems are presented as three main contributions of this thesis, which exercise this framework to improve their performances.

Starting with the object tracking task, the framework is utilized to extract object's inner structure. The algorithm makes use of this inner structure to support a discriminative learning process for mitigating the classic error accumulation effect raising in numerous trackers. In this way, the tracking task is formulated as a prior regularized semi-supervised learning problem. To solve this particular problem, a multi-objective optimization approach is developed. The experiment conducted by the author demonstrates that this tracking algorithm advances state-of-the-art performance of object tracking.

Next, the background subtraction task is studied. In this algorithm, the background is represented by a probabilistic topic model to deal with the dynamic background challenge. This topic model takes advantage of the framework to control topic

proportions, which is shown a good descriptor for recurring pixel movement in dynamic background. In order to make the topic model suitable for this on-line task, an incremental learning approach is designed. In the experiment, this background subtraction algorithm outperforms the alternatives in challenging benchmarks.

Finally, the proposed framework is expanded by applying it on a single image processing task, airborne ship detection. The algorithm handles this detection problem by modeling the ocean background and treating the ship pixels as outliers. For simultaneously encoding the dynamic nature and the local similarity of ocean background texture, the framework is used to explore the majority of pixel intensity across the image plane. An extensive experiment shows robustness and accuracy of the ship detection algorithm on a large number of tested images.

Acknowledgements

I would like to take this opportunity to thank some very special people. First of all, I would like to thank my supervisor Professor T. Kirubarajan, for accepting me as a Ph.D. student, for believing in me and for his expert advice and guidance during my graduate studies. It has been a great privilege and invaluable experience to work with him.

Next, I would like to thank Dr. R. Tharmarasa for much needed advice. I also want to express my gratitude to Professor A. Jeremic and Professor T. R. Field, who assisted me with valuable feedback on my research works. I would like to thank some colleagues in estimation, tracking and fusion research laboratory, Qingsong, Xin, Xue, Yinghui, Keqi, Dan and Krishanth. My sincere thanks goes to the graduate administrative assistant Cheryl Gies. I am also very thankful for the funding from ECE department and the International Excellence Award from the Graduate School of Studies.

Last but by no means least, I would like to thank my mother and father for their unconditional love and consistent support over all these years, without whom all my achievements would have simply been impossible.

Abbreviations

GMM	Gaussian Mixture Model
SAR	Synthetic Aperture Radar
HOG	Histogram of Oriented Gradients
ROI	Region of Interest
SVM	Support Vector Machine

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Extracting useful information from image and video is a fundamental task in computer vision. In this thesis, the spatial relationship among different appearance components on image plane is studied. To discover this relationship, a simple but effective framework based on a grid of image-domain label-distribution is proposed. By cooperating with various appearance models, an universal improvement is demonstrated for a number of distinct computer vision tasks.

Traditionally, how to robustly extract the feature and how to build suitable learning method are the two commonly appearing modules in determining the performance of a vision system. The proposed framework offers a third module: modelling the distribution of each appearance component over image plane. The underlying insight is that a consistency of the label-distributions widely exists spatially or temporally in computer vision tasks. Particularly, an image region of interest is first discretized into a grid of observed visual units, whose size vary from a pixel to a patch depending on different tasks. For modelling the feature of each unit, the framework has no constraint on the appearance module, which makes it compatible with most of the

existing learning methods. As target area is divided into multiple components, the region of interest will be represented by multiple classes. The essential function of this framework is to estimate the occurrence probability of each class on every grid position, which provides a certainty for the output of appearance module for further analysis. Taking advantage of this certainty, a more suitable objective can be formulated compared with those methods only employing appearance information. Having been applied to three different vision tasks, namely object tracking, background subtraction and ship detection, this framework demonstrates its usefulness.

From pattern recognition perspective, most classification and clustering approaches show weakness on representing the highly nonlinear relationship between feature elements. In order to deal with restrained tasks, such as face recognition and pedestrian detection, enlarging the training data by utilizing advanced learning approach would help. But, for those unrestrained tasks, such as arbitrary object tracking and scene understanding, researchers tend to pay more attention on feature engineering, for example, bag-of-words [114] and autoencoder of deep network [48]. Unfortunately, valuable image-domain spatial information has been abandoned through this process. The proposed framework is motivated to keep this useful information.

As shown in Figure.1.1, an image region of interest can be defined as an area within the bounding box, such as a patch extracted from the image or the whole image plane. For the visual units extracted, any supervised or unsupervised multi-class learning routine can be followed. The probability of occurrence on each vertex is modeled by a multinomial distribution, where each element of the parameter corresponds to one class label. For every multinomial distribution on the grid, its conjugate prior, Dirichlet distribution, is utilized. Through Bayesian analysis, the parameter update

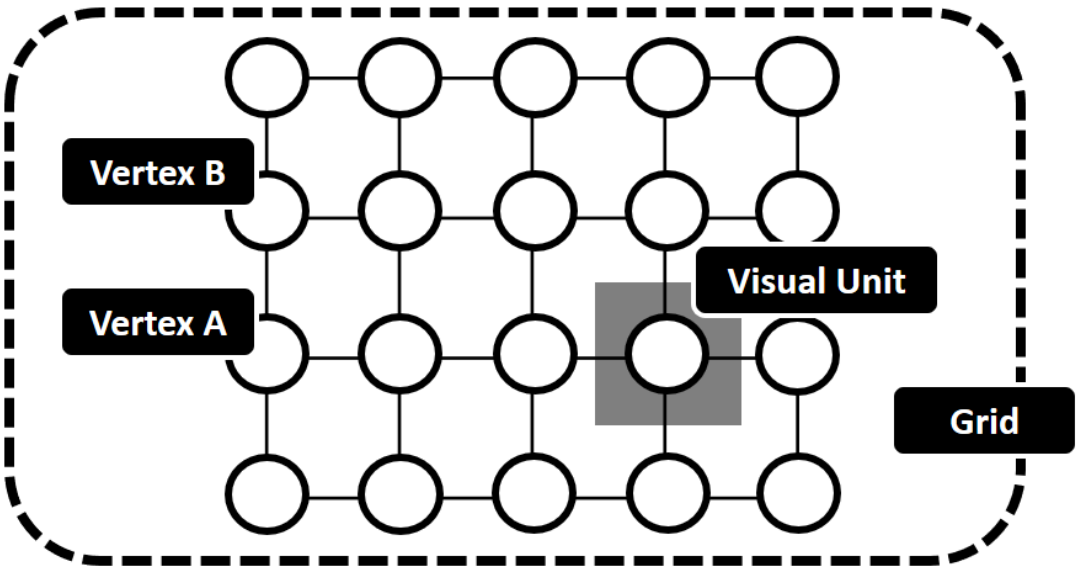

Figure 1.1: Introduction to the structure of the proposed framework

can be formulated as a counting process. Each element of the parameter vector denotes the times of the corresponding class appears in this position, while the prior stands for a pseudo number. In this way, the consistency through time domain, that the probability of certain label occurring on certain vertex will tend to be stable in a period, can be maintained by an on-line updating process. For some tasks, the consistency through image plane, the similarity between adjoining vertexes on the grid, also needs to be considered. In this condition, multiple spatial distributions over the two-dimension grid plane can be applied, whose probabilities at each vertex can be then formulated into the prior parameter vector. For different tasks, this consistency can be fused into the objective function of corresponding task by different ways, which will be presented in the following chapters.

To handle a certain computer vision task, the proposed framework usually encounters two issues. One is determinating the size of a visual unit. Usually, higher level

framework	object tracking	background subtraction	ship detection
grid	structure model	block	image plane
vertex	vertex	pixel	pixel
visual unit	patch	evidence	intensity
label distribution	Dirichlet-multinomial distribution	topic proportion	weight

Table 1.1: The correspondence relationships of terminologies in different algorithms

computer vision task needs a larger patch. For advanced classifiers, such as SVM, neural network and Random Forest, complex feature tends to be used. For generative model, which is often applied in the task formulated as clustering problem, shorter feature is usually used, since there will be a significant computational increase as the feature's dimension raise. The second issue is the class/cluster count. Although most modern classifiers support multi-class, setting down the number of class will still be a problem.

To demonstrate the usefulness of this framework, three novel algorithms handling various computer vision tasks, object tracking, background subtraction and ship detection respectively are proposed in this thesis. Each of the following three chapters will focus on one of them. It is worth to note that, although the proposed framework is the inner-connection between these algorithms, every method has its own contribution to the corresponding computer vision task. For each task, different theory formulation will be introduced. In order to keep a coherence for each theory, author would use different term to present this proposed framework, since it is treated from different perspectives. Although it will not be difficult for readers to recognize it from the context, it would be convenient to briefly introduce these tasks and discuss how the framework plays its role. The correspondence relationships of the terminologies in different algorithms are shown in Table 1.1.

1.1 Discover Target Structure for Object Tracking

Visual object tracking is a fundamental task in computer vision, which has a large number of applications, such as visual surveillance, human computer interaction and video compression. For example, it has been successfully utilized in residential area surveillance system, W^4 system [45]. Actually, in almost any video analysis system, tracking plays its role.

A complete visual tracking system usually includes five modules: initialization, appearance modeling, motion estimation, localization and updating. The initialization phase can be done manually or automatically. Manual way is operated by users to annotate target's location parameters, usually a bounding box or an ellipse on the image plane. The automatic way provides the same information but uses the output of an object detector. Appearance modeling has been demonstrated as the most important part of a tracking system. It focuses on two aspects, designing robust target descriptor and building mathematical model for target recognition. Target descriptors can make use of contour, shape, segmentation, local patch and global region. The local patch and global region are the two mostly used cues, from which various features can be extracted. Recognition approaches branch into two groups, generative and discriminative models. Since there is no commonly agreed technical routine for this module, a clear description can be expressed as attempting to answer two questions, what visual representations are appropriate and robust for object tracking and what types of learning schemes are suitable for visual tracking. Usually, motion estimation task is formulated as a dynamic state estimation problem in literature, which can be completed by taking advantage of linear regression [82], Kalman filters [27] or particle filters [6]. In recent years, researchers tend to spend less effort on

this module. The localization task is often performed by a greedy search or a maximum posterior estimation process. The goal is to determine the target state, such as the position on image plane, based on the observation on current frame. Although some trackers do not operate update, most update in order to handle variation in appearance. Generally, one of the old samples is replaced by the last seen for update.

Despite a large number of researchers are attracted to this topic, robust tracking still suffers from various challenges. One is low-quality camera, which consequentially causes low frame rate, low resolution, low bit-depth and color distortion. Another is challenging object, such as nonrigid object, small-size object, out-of-plane rotation and complicated pose. Challenging environment is also commonly seen, including illumination change, rapid camera motion, full occlusion and noise disturbance. For real-world setting, researchers also need to consider real-time processing requirement.

The proposed framework is utilized to explore object inner structure. In this task, the grid is constructed from the input bounding box of object intended to be tracked and each visual unit is set as a 16×16 patch extracted from the vertex. All the visual samples are fed into the Random Forest classifier. Different from typical appearance model, the set of label distributions of the framework can provide additional certainty for the classification result. Since it shows the probability of a certain appearance class occurring on certain positions, they essentially represent the structure of object. As this structure tends to hold consistently in a period, this novel pattern can be applied to support target tracking. Furthermore, based on this framework, the tracking task is re-formulated as a semi-supervised learning problem, where the generative structure is used to guide the discriminative learning. This pattern has also been utilized in localization and updating phases. To determine target position, a grid of certainties

is used for measurement in a greedy search mechanism. Traditional updating tends to absorb noisy samples into model, which certainly deteriorates the tracker and even causes drift. Given this additional probabilistic measure, only credible samples can be updated.

1.2 Discover Motion Pattern for Background Subtraction

Background subtraction is usually applied as the first step of many computer vision systems to detect moving objects in a video sequence without any prior knowledge. Its major application is the video surveillance system, where persons, vehicles and animals are first detected before they operate more complex modules for intrusion detection, tracking and individual counting.

The core of a background subtraction algorithm is creating its own background model. In the simplest way, this can be obtained by manually setting a static image from the sequence, which represents the background and contains no moving object. Then, the absolute difference between the current frame and the static image can be computed. However, this is not the best choice. Any change of background scene would cause foreground segmentation failure. For improvement, the previous frame rather than a static image can be used, which introduces an important concept, background maintenance. Alternatively, [57] suggests the initialization and maintenance phases of the background model by the arithmetic mean of the pixel between successive frames. Taking advantage of the same methodology, statistic techniques are

employed to solve this task. The most well-known method is a parametric probabilistic background model proposed by Stauffer and Grimson [103], and improved by Hayman and Eklundh [113]. In this algorithm, distributions of each pixel color is represented by a sum of weighted Gaussian distributions defined in a given colorspace, the Gaussian mixture mode.

An ideal environment for background subtraction contains three factors, fixed camera, static background and constant illumination. In practice, however, a number of challenging situations may appear. 1) noisy image: these sequences usually come from a poor-quality source; 2) camera jitter: wind may cause the camera move; 3) automatic adjustment: this camera function might automatically adjust the pixel value representing the same color between different frames; 4) illumination change: it could be either gradual such as outdoor scene or sudden such as light switch; 5) bootstrapping: in the training period, there is no static frame available; 6) camouflage: foreground pixels show similarity with the background, which makes them hard to distinguish. 7) foreground aperture: the moving object has a uniform colored region, making the changes inside the region hard to be detected; 8) dynamic background: although background objects may move, these objects should not be treated as foreground, like trees; 9) sleeping object: foreground object might become motionless after going into the scene; 10) shadow: though shadow does not belong to the region of interest for most of applications, they are often detected as foreground.

To handle the background subtraction task, the proposed framework is utilized to encode the recurring movement of the pixels, which improves system performance when the first two ideal environmental factors are not met. Each frame of the input sequence is divided into multiple independent equal-size blocks, each of which is

masked by the grid. A visual unit is set as one pixel. The intensities appearing on the block are modeled by multiple Gaussian distributions, where different Gaussian components intend to represent different objects. Since some objects of background might have little movement as mentioned such as tree and fountain, the probability of each Gaussian component appearing on a certain position of the grid should be modeled. In the algorithm, the proposed framework is utilized to represent this probability. The reason of this design for the algorithm is that the recurring movement can be summarized as a pattern, which makes the probabilities of the label distributions consistent in the time domain. By fusing this framework with the Gaussian models, background subtraction task can be represented by a probabilistic topic model.

1.3 Discover Dynamic Background for Ship Detection

Recently, ship detection has attracted wide interest for maritime security and other applications, such as traffic control, protection against illegal fisheries, oil discharge control and pollution monitoring. Cargo monitoring from airborne images covers a large enough ocean scene, which consequently can achieve a continuous monitoring of targets' locations and movements. However, it usually suffers two issues, low-contrast scene and high computational cost.

The topic of ship detection has been extensively studied for decades. But few research works directly attempt to handle the scene from the airborne view. Most related literature shows two tendencies. One is detecting the ships from space, where the image naturally has a higher resolution than other remote sensing images and

the view of each object is highly restrained as "top-down". For instance, [22] utilized neural network to classify small ships and built a complete system for ship detection. A major weakness of these methods come from the negative influence from clouds and at night. The other is that methods tend to focus on a particular sensor. For example, a number of ship detection methods have been proposed based on the synthetic aperture radar (SAR). Most of them employ a constant false-alarm rate detector with a certain background distribution [32][113][25]. For comprehensive review of ship detection in SAR images, readers are referred to [138]. Other methods include detecting long ship tracks in Advanced Very High Resolution Radiometer imagery [74][124] and in airborne infrared images [28][112][29]. Generally, the literature about ship detection shows no convergence on a major technical routine.

Taking advantage of proposed framework, a novel ship detection algorithm is presented, in which the detection task is treated as a salient object extraction problem by modeling the ocean background. The major difficulty is that the algorithm has to effectively detect the salient pixels in a significantly low-contrast image. In this algorithm, the proposed framework plays a key role, that the characteristic of ocean background is encoded into a statistical model and each ship pixel is considered as an outlier. The grid is masked on the whole image plane. Similar to the background subtraction, each visual unit is a pixel, whose value is drawn from a Gaussian mixture model. Each weight of Gaussian component is also expressed by the multinomial distribution. Different from the fore-mentioned two algorithms, the input of this algorithm is a single image rather than a video, which can not accumulate label's image-domain distribution. The consistency across the image plane is explored by a set of spatial distributions over the image plane, whose details will be discussed in the

later chapter. The spatial distribution is fused into proposed framework by treating it as a hyper-distribution. Particularly, each element of the prior parameter vector of Dirichlet distribution equals to the probability of the corresponding vertex on the grid of the spatial distribution. An Experiments on a number of datasets shows the ocean background is well modeled by this proposed algorithm.

Chapter 2

Robust Discriminative Tracking via Structured Prior Regularization

In this chapter, author addresses the problem of tracking an object in a video sequence given its location in the first frame and no other information. Many existing discriminative tracking algorithms usually train a classifier in an on-line manner to separate the object of interest from the background. Slight inaccuracies in the tracking may result in incorrectly labelled training set, which can degrade the tracker. Although a number of approaches such as semi-supervised learning and multiple instance learning have been developed to address this problem, some critical issues still remain unsolved. This chapter aims to mitigate the shortcomings by exploiting a reliable generative model to support the discriminative learning process. A prior model based on a set of structured Dirichlet-multinomial distributions is proposed to preserve the target's structure information. This prior is then formulated as a regularization term in a training objective function, which casts the tracking task as a prior regularized semi-supervised learning problem. A multi-objective optimization

method is developed to search for the solution, taking advantage of a decision maker inside to control the conflicts between different modules. The experiments show that this proposed method outperforms standard algorithms on challenging datasets. It is also demonstrated that the algorithm significantly mitigates the error accumulation effect.

2.1 Introduction

Object tracking is a fundamental task in computer vision, which can be used in numerous different applications, including surveillance, autonomous vehicles, intelligent robots, augmented reality, and medical imaging. Although many approaches have been proposed in the literature, robust tracking still remains a challenging problem [131]. In real-world settings, objects are typically complex and difficult to track due to pose, rotation, illumination, blur, occlusion, abrupt motion and background clutter. Effectively modelling and maintaining target appearance are of prime importance for the success of a tracking algorithm, which has attracted much attention in recent years [117] [55] [85] [2] [60] [56] [125] [94] [11].

Typical discriminative tracking algorithms [9] [7] consist of a two-step "estimate-update" process [40]. First, a predefined appearance model is utilized to evaluate the likelihood in the current frame and a decision strategy is used to estimate the target's image position. Second, once the position is determined, new foreground/background information is extracted to update the appearance model. A critical drawback of this mechanism is that the sub-optimal estimation caused by noise will lead to inaccuracies in the update step, consequently degrading the model. An inaccurate model in turn may result in even worse estimation results [9]. Although some methods like multiple

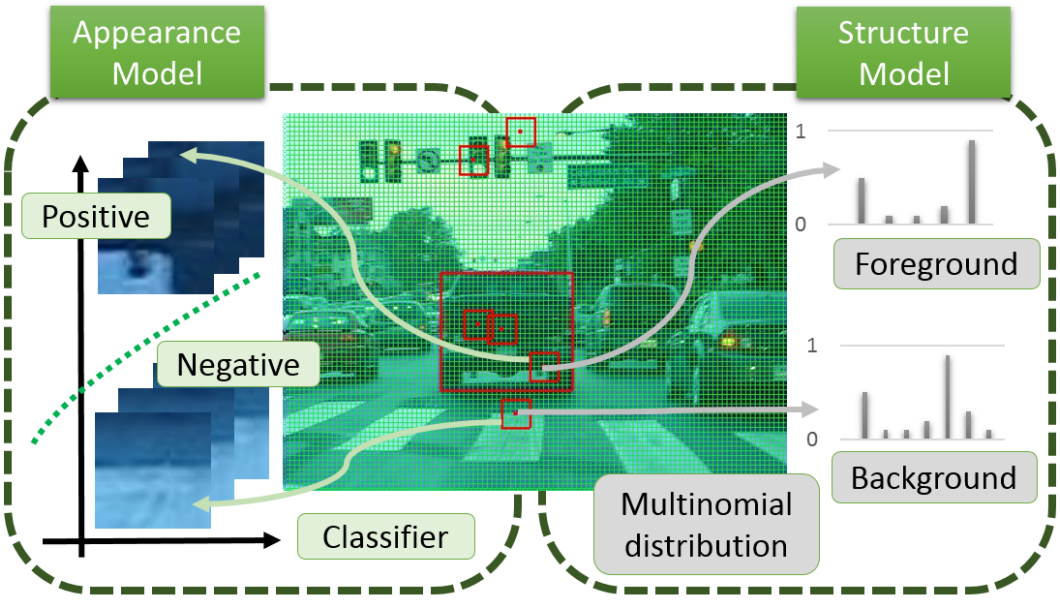

Figure 2.1: Left side: discriminative appearance model; Right side: generative structure model (image source: [127]).

instance learning [9] [134] and semi-supervised learning [40] [102] have been proposed, accurate object tracking still remains a challenging issue.

In this chapter, our motivation is to devise a new strategy based on reliable rules to update the model that minimizes error accumulation. The goal is to improve visual tracking robustness, especially to avoid drift. The key idea is that the model should adapt itself over the video sequence rather than explicitly retraining the classifier based on noisy features as supervised learning. To achieve this, a generative model consisting of a grid of structured Dirichlet-multinomial distributions is proposed. In this model the target and the background are represented by grid structures as shown in Figure 2.1, where patches extracted on these vertices are utilized to generate training features assigned with multiple labels to feed a Random Forest classifier. A

Dirichlet-multinomial distribution fixed on each vertex is used to maintain corresponding labels confidence. By formulating the generative model as an additional regularization term in the objective function, the tracking task is cast into a prior regularized semi-supervised problem. The algorithm treats current patches as unlabelled data and learns their labels in a semi-supervised manner with structure prior and appearance prior, which may have conflicts between them. To address this conflict, a multi-objective optimization technique is proposed. The experimental results show that the proposed model significantly improves the update phase to minimize drift over time, which consequently yields better tracking results.

The main contributions of this chapter are: 1) Visual tracking task is cast as a prior regularized semi-supervised problem; 2) A multi-objective optimization algorithm is proposed to solve this regularized semi-supervised problem; 3) A novel patch-based grid target representation is designed; and 4) A new adaptive Random Forest is proposed to speed up the optimization process. The proposed algorithm is called the structured prior regularization tracker (SPRT).

2.2 Related Work

The related works are grouped into three categories.

2.2.1 Discriminative Appearance

A major challenge in discriminative tracking is how to update the adaptive appearance model [9]. This is usually done by taking the current tracker location as the positive feature and sampling the negative ones around [131] [39] [71] [21] [53] [63]. But, the

process usually causes the model to be updated with noisy and potentially misaligned features, which often leads to the drift problem. To alleviate this, an on-line semi-supervised boosting method is proposed in [40], where labelled features come from the first frame only and subsequent training features are left unlabelled. Similarly, a multiple instance learning technique is applied in [9] with the key assumption that a negative bag consists of all negative instances whereas a positive bag contains at least one positive instance. In [52], a semi-supervised-like approach, in which positive and negative features are selected via an online classifier with structural constraints is proposed. Other approaches to address drift include explicitly detecting tracking failures and occlusions with heuristics [139] and maintaining a pool of trackers [133]. In [120], the authors study each component of the tracking system and identify some important factors for good performance. Recently, significant advances in applying convolutional neural networks on the visual tracking problem have been made. A deep learning tracker using a multi-layer autoencoder network is proposed in [119], where the network is pre-trained in an unsupervised manner. In [64], a two-layer network based on binary features is trained by the tracker without requiring a pre-training procedure.

2.2.2 Tracking Based on Multiple Patches

Patch-based approaches have the added benefit of robustness to occlusion [56] [129] [8] [10]. Numerous methods have been proposed to exploit this advantage. In [2], a tracking method using multiple image fragments is presented to handle occlusion and pose changes, where every fragment votes on the possible position and scales of the

object. In [85], the constantly changing foreground shape is modelled based on multiple rectangular blocks to track articulated objects with large variation in appearance and shape, where the blocks' positions in the tracking window are adaptively estimated. In [51], the target is represented by a set of small patches on a regular grid and l_1-sparsity is applied to search for the final estimate. The patch-based model can also be utilized along with Random Forest [16] to handle the detection, recognition and tracking of objects [87]. In [56], a flexible patch-based method is designed to track the object whose geometric appearance is drastically changing over time. By employing Basin-Hopping sampling, this method can efficiently find an optimal state. In [111], a set of patches on a grid structure is tracked by a flock of trackers that is robust to certain drifts. Recently, the correlation filter based method (KCF)[47] was proven to efficiently yield robust tracking performance, taking advantage of convolution theorem to effectively learn the object template. There is another tracker in [66], also exploiting local context, where the tracked parts are automatically selected by sampling and a reliable part is defined as being trackable and sticking on the target. In [88], a method for detecting deformable objects, whose object model is composed of a regular grid of small parts with their locations depending on the distance from neighboring parts is proposed. In [109], sets of pixels/patches are used to model and track the target. A technique based on solving linear decomposition problems with enforced sparsity is proposed for matching. In these patch based trackers, grid representation, star model and tree structure are the three most commonly used structures for representing targets [136].

2.2.3 Semi-supervised Random Forest

Semi-supervised learning has received significant interest in the literature. A commonly used discriminative method is the "cluster-based" one, which aims to place the decision boundary in low density regions directly [23]. For example, in the entropy regularization method [41], an additional term that minimizes the entropy of the label distribution predicted on unlabelled data is employed to augment the traditional conditional label likelihood objective function. In [77], a similar expectation regularization method is proposed to achieve semi-supervised learning while incorporating prior information about the class distribution into discriminative training to improve performance. Recent references on semi-supervised learning can be found in [18] [50].

Random Forest [16] is an ensemble learning technique, which has been successfully applied in many computer vision and machine learning applications. It was demonstrated to be better or at least comparable to other discriminative modelling techniques in detection [31], object recognition [87], semantic segmentation [99], real-time key point recognition [62] and tracking [53]. For a comprehensive introduction to Random Forest, the reader is referred to [24]. The semi-supervised solutions for the Random Forest have also been proposed in [24][61]. The first work introduces a simple and efficient semi-supervised Random Forest via transductive learning while the latter one employs the deterministic annealing technique to iteratively achieve semi-supervised learning.

The original Random Forest was proposed in [5], and consists of ensembles of multiple independent decision trees. A decision tree is a (binary) tree-structured classifier that makes a prediction by routing a feature $\boldsymbol{x} \in \mathbb{X}$ in the feature space $\mathbb{X}$ through the root to a leaf, where classification takes place. A label-distribution is stored at

each leaf $y \in \mathbb{Y}$ that could be associated to any feature, where $\mathbb{Y} = (0, 1, ..., k, ..., K)$ denotes the label domain. At each internal node, a feature is forwarded to the left or to the right based on the output of a node-specific split function. The Random Forest makes a prediction about a feature by combining single response collected from every tree, and the final confidence is defined as $c(k|\boldsymbol{x}) = 1/V \sum_{v=1}^{V} c_v(k|\boldsymbol{x})$ where $c_v(k|\boldsymbol{x})$ is the estimated confidence of class labels on the leaf of the v-th tree. The final decision is defined as $f(x) = \arg\max_k c(k|x)$.

In the training phase, every tree is trained independently [24]. This procedure is carried out recursively, where an internal node is grown with an associated split function selected from a randomly generated parameter set to maximize the expected information gain (2.17) about the label distribution due to the split of the training data into two sets, left and right ones $\boldsymbol{D} = {}^{l}\boldsymbol{D} \cup {}^{r}\boldsymbol{D}$. The node's left and right subtrees are recursively grown with their own training data ${}^{l}\boldsymbol{D}$ and ${}^{r}\boldsymbol{D}$, respectively, until some terminal criterion, e.g., minimum size of data set, maximum depth or entropy threshold, is met. After this, the confidence of class is estimated on each leaf.

2.3 Algorithm

The proposed algorithm consists of three modules, the appearance model, the structure model, and the optimization algorithm. The appearance model is responsible for preserving similarity in appearance by maintaining a built-in semi-supervised Random Forest classifier, whose loss function is denoted as $\boldsymbol{L_a}$. The structure model is used to localize the target by identifying the most similar structure pattern, and its loss function is denoted as $\boldsymbol{L_s}$. Based on these two models, the tracking task can be cast into a prior regularized semi-supervised learning problem, whose objective

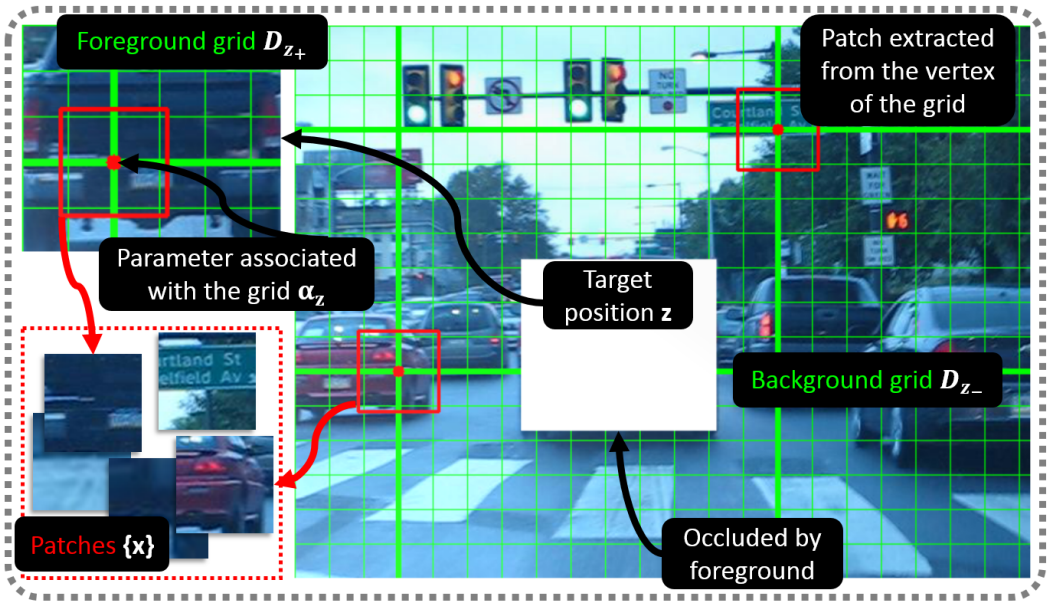

Figure 2.2: Representation: Grid, the green fixed-structure lattice masked on the target or the background with 3 pixels between adjacent points/crosses; Vertex, the red points/crosses at both foreground and background grids; Patch, the red rectangle centered on corresponding vertex (image source: [127]).

function is defined as $\boldsymbol{L_s} + \boldsymbol{L_a}$. A multi-objective optimization algorithm is proposed to find the optimal solution for the objective function.

2.3.1 Representation

As shown in Figure 2.2, the object (foreground) and the background are represented as two fixed-structure grids. On each vertex of a grid, a 16×16 local image patch $\boldsymbol{x}$ is extracted to model the target's partial appearance and to feed the classifier. A Dirichlet-multinomial distribution parameterized by $\boldsymbol{\alpha}$ is also associated with the vertex to maintain the confidence of the classification result for the corresponding patch.

Two sets of patches are obtained from the foreground and background separately,

where, for each set, patches are grouped into multiple classes K^+ and K^- rather than one. In other words, our foreground (or background) would be represented by more than one label in order to express the target's inner-structure, as shown in Figure 2.4(b)–(d).

Given the labels, the foreground grid associated with the set of structured distributions will be employed to estimate target location $\boldsymbol{z}$. A dependent relationship holds such that once the grid settles, the position of each vertex is determined, since the grid structure is fixed and rigid. Thus, the position of each distribution will not be explicitly indicated in this chapter, but only be presented by $\boldsymbol{\alpha}_{\boldsymbol{z}}$ illustrating the corresponding distribution parameter $\boldsymbol{\alpha}$ as the grid localizing at $\boldsymbol{z}$ (see Figure 2.2).

2.3.2 Appearance Model

To preserve continuity in appearance between the current frame and the previous ones, semi-supervised Random Forest is applied in this model. The goal is to estimate labels of new patches and a new classifier parametrized by $\boldsymbol{\theta}^{t+1}$ constrained by previous data $\boldsymbol{D}^t = \{\boldsymbol{x}^t, y^t\}$ and current patches $\{\boldsymbol{x}^{t+1}\}$.[2] The loss function can be cast into a semi-supervised learning form as follows:

$$\boldsymbol{L_a} = \sum_{\boldsymbol{D}^t} l\left(y^t, f(\boldsymbol{x}^t; \boldsymbol{\theta}^{t+1})\right) + \sum_{\boldsymbol{D}^{t+1}} l\left(y^{t+1}, f(\boldsymbol{x}^{t+1}; \boldsymbol{\theta}^{t+1})\right) \tag{2.1}$$

where function $l(y, f(\boldsymbol{x}; \boldsymbol{\theta}))$ is the Random Forest's loss function, $\boldsymbol{D}^{t+1} = \{\boldsymbol{x}^{t+1}, y^{t+1}\}$ and $f(\boldsymbol{x}; \boldsymbol{\theta})$ is the deterministic function of trained classifier with parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, where $\boldsymbol{\Theta}$ is the Random Forest's parameter domain.

In the Random Forest, a classification margin for a data item pair $(\boldsymbol{x}, y)$ is defined

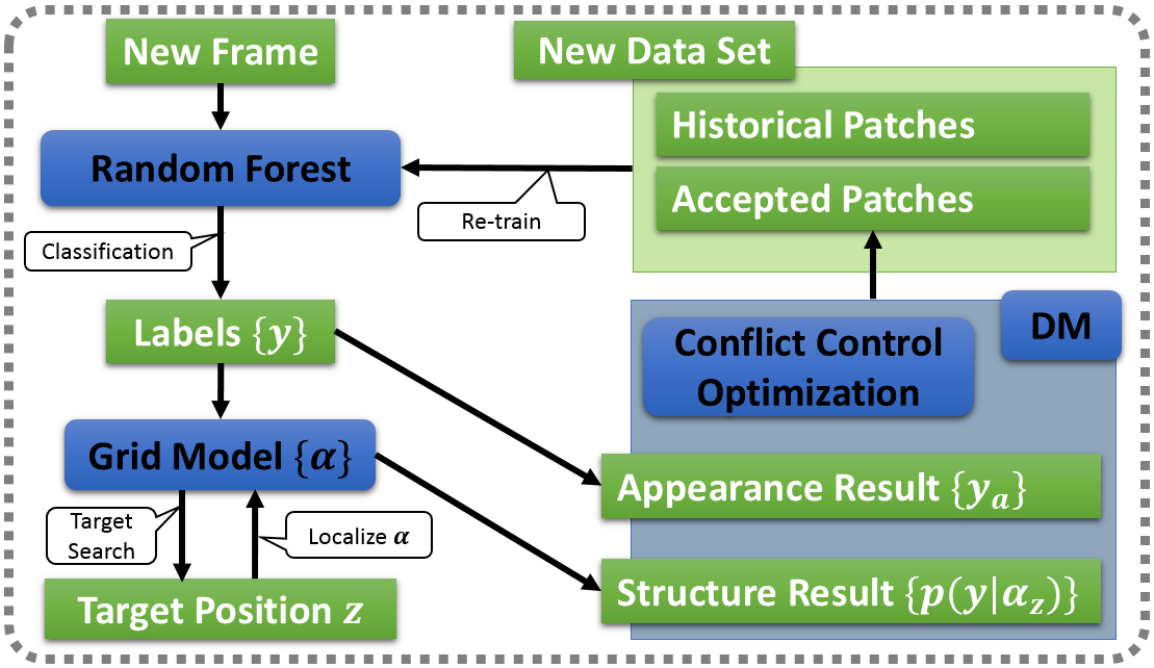

Figure 2.3: Tracking framework

as $c(y|\boldsymbol{x}) - \max_{k \neq y} c(k|\boldsymbol{x})$ in [16], where $k \in \mathbb{Y}$. The training process can be seen as a process that greedily and recursively maximizes this margin [61]. The loss function can be formulated as the negative margin as follows:

$$l(y, f(\boldsymbol{x}; \boldsymbol{\theta})) = \max_{k \neq y} c(k|\boldsymbol{x}) - c(y|\boldsymbol{x}) \tag{2.2}$$

One can make use of the training process to find the optimization solution for the appearance loss. Because of the margin's attribute, the condition that $l(y, f(\boldsymbol{x}; \boldsymbol{\theta})) \in [-1, 1]$ holds. In semi-supervised learning, there is no category information about unlabelled data item at the beginning. Note that label 0 is assigned to the unlabelled features and the author set $l(0, f(\boldsymbol{x}; \boldsymbol{\theta})) = 1$.

2.3.3 Structure Model

The structure model consists of two sets of distributions $\{p(y|\boldsymbol{\alpha}_{\boldsymbol{z}}^t)\}$ localizing on the two grids parametrized by $\{\boldsymbol{\alpha}_{\boldsymbol{z}}^t\}$, whose observations are the labels $\{y\}$. The superscript denotes the $\boldsymbol{\alpha}_{\boldsymbol{z}}^t$ estimated from frame t. The loss function is formulated as a negative sum of all the probabilities on the foreground grid, given by

$$\boldsymbol{L_s} = -\sum_{\boldsymbol{D}_{\boldsymbol{z}_+}^{t+1}} p(y^{t+1}|\boldsymbol{\alpha}_{\boldsymbol{z}}^t) \tag{2.3}$$

where $\boldsymbol{D}_{\boldsymbol{z}_+}^{t+1}$ denotes the labels and patches covered by the foreground grid localizing position $\boldsymbol{z} \in \mathbb{Z}$, $\mathbb{Z}$ is the image domain, as shown in Figure 2.2, $\boldsymbol{D}_{\boldsymbol{z}_-}^{t+1}$ indicates the background ones and $\boldsymbol{D}^{t+1} = \boldsymbol{D}_{\boldsymbol{z}_+}^{t+1} \cup \boldsymbol{D}_{\boldsymbol{z}_-}^{t+1}$. Note that, although $\boldsymbol{D}_{\boldsymbol{z}_+}^{t+1}$ includes both labels and patches of the dataset, only the labels are utilized in the structure model.

Since the grid's structure is invariant, once a candidate position $\boldsymbol{z}$ is fixed, the observation labels $\boldsymbol{D}_{\boldsymbol{z}_+}^{t+1}$ and corresponding parameters $\{\boldsymbol{\alpha}_{\boldsymbol{z}}^t\}$ on this grid are determined accordingly. In this model, all Dirichlet-multinomial distributions are assumed independent of each other. A sum rather than multiplication is employed here because there may be background labels observed in foreground grid with probability 0, which can make the multiplication of probabilities an inappropriate measure to quantify the match between the model and the observation. The loss is modelled by this formula because it illustrates the quality of representing target's structure. For instance, if the change in observed labels is small in two successive frames, meaning that the appearance is modelled appropriately, the structure loss should also tend to be small, which could be expressed by the function.

The loss function can be used in two steps. First, it is utilized to estimate the target location $\boldsymbol{z}$ by minimizing $\boldsymbol{L_s}$ via sliding the foreground grid through all candidate positions, shown by the "Target-Search" black arrow in Figure 2.3 and formulated as

$$\boldsymbol{z} = \arg\max_{\hat{\boldsymbol{z}} \in \mathbb{Z}} \sum_{\boldsymbol{D}^{t+1}_{\hat{\boldsymbol{z}}_+}} p(y_a^{t+1} | \boldsymbol{\alpha}^t_{\hat{\boldsymbol{z}}}) \tag{2.4}$$

where $\hat{\cdot}$ denotes the variable to be estimated. In this step, the observations $\{y_a^{t+1}\}$ are known and given by the classification. The symbol y_a^{t+1} indicates that the label on current frame $t+1$ is estimated from the appearance model. Second, it is utilized to align each distribution $\boldsymbol{\alpha}^t_{\boldsymbol{z}}$ with its observation and provide a probabilistic measure $p(y|\boldsymbol{\alpha}^t_{\boldsymbol{z}})$, based on the estimated location $\boldsymbol{z}$ from the previous step, shown by the "Localize $\boldsymbol{\alpha}$" arrow in Figure 2.3. The difference between these two steps is that the first one globally focuses on estimating grid location while the second partially measures the confidence of each label. Besides, if we continue to minimize the loss function $\boldsymbol{L_s}$ over y, another set of labels from structure side $\{y_s^{t+1}\}$ can also be estimated as

$$\{y_s^{t+1}\} = \arg\max_{\hat{y}^{t+1} \in \mathbb{Y}} \sum_{\boldsymbol{D}^{t+1}_{\boldsymbol{z}_+}} p(\hat{y}^{t+1} | \boldsymbol{\alpha}^t_{\boldsymbol{z}}) \tag{2.5}$$

The two label sets $\{y_a^{t+1}\}$ and $\{y_s^{t+1}\}$ may not be the same, which explicitly indicates conflict between appearance and structure.

Dirichlet-Multinomial Distribution

The Dirichlet-Multinomial [86] model is expressed as two terms, the Dirichlet prior $Dir(\boldsymbol{\alpha}^0)$ and the multinomial distribution $Mult(\boldsymbol{\phi})$, where $\boldsymbol{\alpha}^0$ and $\boldsymbol{\phi}$ are parameters

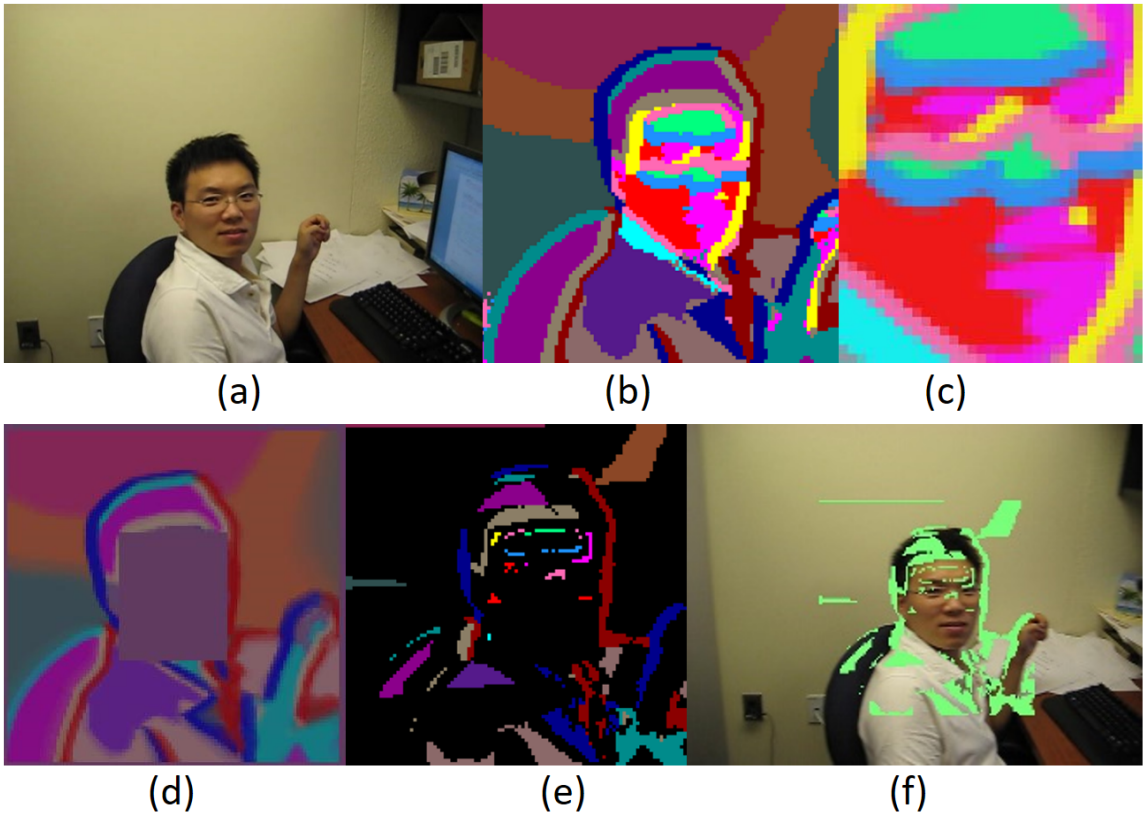

Figure 2.4: Intermediate result:(a) original frame; (b) labels of the whole frame (classification results); (c) visualization of distributions on foreground grid; (d) visualization of distribution on background grid; (e) accepted patches and their corresponding labels; (f) accepted patches (image source: [127]).

for the two distributions, respectively. Then,

$$\begin{aligned} \boldsymbol{\phi} &\sim Dir(\boldsymbol{\alpha}^0) \\ \{y\}^{1:t} &\sim Mult(\boldsymbol{\phi}) \end{aligned} \tag{2.6}$$

Based on the conjugate relationship between Dirichlet and multinomial, given a sequence of observations $\{y\}^{1:t}$ on a vertex, one has

$$p(\boldsymbol{\phi}|\{y\}^{1:t}, \boldsymbol{\alpha}^0) = Dir(\boldsymbol{\alpha}^t) \tag{2.7}$$

where $\boldsymbol{\alpha}^t = (\alpha_1^t, \alpha_2^t, \alpha_3^t, ..., \alpha_{K^{(+/-)}}^t)$, K^+ is the number of classes for the patches in the foreground, $K = K^+ + K^-$, and $\alpha_k^t = \alpha_k^0 + \sum_{\{y\}^{1:t}} \mathbb{1}\{y = k\}$. Here, the k-th element of $\boldsymbol{\alpha}^t$ equals to the corresponding element in $\boldsymbol{\alpha}^0$ plus the number of times class k is observed in $\{y\}^{1:t}$. From the result, it can be seen that the Dirichlet parameters $\boldsymbol{\alpha}^0$ can be regarded as "pseudo-counts" of the labels.

In the tracking phase, the goal is to find the probability of a label centered on a vertex belonging to a particular class k, given a sequence of previous labels of this vertex. Based on (2.7), one has

$$\begin{aligned} & p(y^{t+1}{=}k|\{y\}^{1:t}, \boldsymbol{\alpha}^0) \\ & \quad = \int p(y^{t+1}{=}k|\boldsymbol{\phi}) p(\boldsymbol{\phi}|\{y\}^{1:t}, \boldsymbol{\alpha}^0) d\boldsymbol{\phi} \\ & \quad = \frac{\alpha_k^t}{\sum_{j=1}^{K^{(+/-)}} \alpha_j^t} \end{aligned} \tag{2.8}$$

The probability in (2.8) is a percentage of the total number of prior "pseudo-counts" plus observation counts on the class. Thus, only $\boldsymbol{\alpha}^t$ needs to be stored in the implementation. Because of the fixed grid structure, the parameter $\boldsymbol{\alpha}^t$ has to be localized based on the foreground grid location $\boldsymbol{z}$ before being utilized, i.e., the parameter depends on $\boldsymbol{z}$. Each generative model is denoted as $p(y|\boldsymbol{\alpha}_{\boldsymbol{z}}^t) = p(y|\{y\}^{1:t}, \boldsymbol{\alpha}^0)$. If a background label is observed in the foreground grid, the probability is set to 0.

2.3.4 Optimization

The complete objective function is obtained by making use of the structure loss function as a regularization term added to the appearance loss function. This structured

prior regularized semi-supervised learning problem can be formulated as follows:

$$\begin{aligned}\left(\{y^{t+1}\}, \boldsymbol{\theta}^{t+1}, \boldsymbol{z}\right) = \underset{\substack{\hat{y}^{t+1} \in \mathbb{Y} \\ \hat{\boldsymbol{\theta}}^{t+1} \in \boldsymbol{\Theta} \\ \hat{\boldsymbol{z}} \in \mathbb{Z}}}{\arg\min} \Bigg[& \sum_{\boldsymbol{D}^t} l\Big(y^t, f(\boldsymbol{x}^t; \hat{\boldsymbol{\theta}}^{t+1})\Big) \\ & + \sum_{\boldsymbol{D}^{t+1}} l\Big(\hat{y}^{t+1}, f(\boldsymbol{x}^{t+1}; \hat{\boldsymbol{\theta}}^{t+1})\Big) - \sum_{\boldsymbol{D}^{t+1}_{\hat{\boldsymbol{z}}_+}} p\Big(\hat{y}^{t+1} | \boldsymbol{\alpha}^t_{\hat{\boldsymbol{z}}}\Big) \Bigg]\end{aligned} \tag{2.9}$$

where y^{t+1} denotes the label finally determined on the current frame in contrast to y_a^{t+1}, which is an intermediate result. This function aims to classify patches by considering not only the spatial relationship between the current data items and historical ones in feature space as traditional supervised discriminative model, but also the structural constraint of the target represented by a generative model.

Multi-Objective Optimization

Since the appearance loss and the structure loss may conflict with each other, a global multi-objective optimization method shown in Figure 2.3 is proposed. This makes use of a decision maker (DM) to find the Pareto optimal solution. In this framework, optimizing appearance $\boldsymbol{L_a}$ shown as "Random Forest" in Figure 2.3 and structure $\boldsymbol{L_s}$ shown as "Grid model" are separated into two processes. The DM determines what kind of estimated label is acceptable by considering results from both sides.

The main iterative steps are the follows: 1) initializing every label in $\boldsymbol{D}^{t+1}$ to 0; 2) optimizing $\boldsymbol{L_s}$ and $\boldsymbol{L_a}$ separately; 3) asking for the acceptance information from DM; 4) generating of a new Pareto optimal solution according to the acceptance; 5) terminating if DM stopping criterion is met, otherwise going back to 2). The core

algorithm is shown in Algorithm 1.

On the appearance side, the optimization at iteration i becomes

$$\left(\boldsymbol{\theta}^{t+1}\right)^{i} = \underset{\hat{\boldsymbol{\theta}}^{t+1}\in\boldsymbol{\Theta}}{\arg\min} \sum_{(\boldsymbol{D})^{i}} l\left(y, f(\boldsymbol{x}; \hat{\boldsymbol{\theta}}^{t+1})\right) \tag{2.10}$$

where $(\boldsymbol{D})^{i} = \boldsymbol{D}^{t} + (\boldsymbol{D}^{t+1})^{i}$, set $(\boldsymbol{D}^{t+1})^{i}$ denotes accepted labels $\{(y^{t+1})^{i}\}$ and corresponding patches $\{(\boldsymbol{x}^{t+1})^{i}\}$, $(\boldsymbol{\theta}^{t+1})^{i}$ are estimated Random Forest parameters in the i-th iteration. The optimization of the appearance loss function (2.1) is simplified into a classic training process of the Random Forest based on historical data and accepted data. Since unaccepted labels remain 0 and $l(0, f(\boldsymbol{x}, \boldsymbol{\theta})) = 1$, the second term of (2.1) can be separated into a constant and the term formulated by the sum of the loss functions based on accepted data, which can be combined with the first term of (2.1). The process is shown by the "Re-train" arrow in Figure 2.3. As a result, a new label set of appearance for the next iteration $i+1$ can be obtained by executing classification on the current frame, shown by "Classification" arrow in Figure 2.3 and Figure 2.4(b), as

$$\left(y_{a}^{t+1}\right)^{i+1} = f\left(\boldsymbol{x}^{t+1}; \left(\boldsymbol{\theta}^{t+1}\right)^{i}\right) \tag{2.11}$$

On the structure side, a new target location $(\boldsymbol{z})^{i+1}$ is estimated based on $\{(y_{a}^{t+1})^{i+1}\}$. Equation (2.4) can be rewritten as

$$(\boldsymbol{z})^{i+1} = \underset{\hat{\boldsymbol{z}}\in\mathbb{Z}}{\arg\max} \sum_{\left(\boldsymbol{D}_{\hat{\boldsymbol{z}}_{+}}^{t+1}\right)^{i+1}} p\left(\left(y_{a}^{t+1}\right)^{i+1} | \boldsymbol{\alpha}_{\hat{\boldsymbol{z}}}^{t}\right) \tag{2.12}$$

Once the location is estimated, the foreground grid is determined with its distributions $\{p(y|\boldsymbol{\alpha}_{(\boldsymbol{z})^{i+1}}^{t})\}$. If we continue to minimize the structure loss function over labels as in

Algorithm 1 Proposed SPRT Algorithm

Input: image patches $\{\boldsymbol{x}^{t+1}\}$, historical data $\boldsymbol{D}^t$, Random Forest parameters $\boldsymbol{\theta}^t$, Dirichlet-multinomial parameters $\{\boldsymbol{\alpha}^t\}$;
Output: target position $\boldsymbol{z}$, current labels $\{y^{t+1}\}$, new Random Forest parameters $\boldsymbol{\theta}^{t+1}$, new distribution parameters $\{\boldsymbol{\alpha}^{t+1}\}$;

set $i = 1$;
set $(\boldsymbol{\theta}^{t+1})^1 = \boldsymbol{\theta}^t$;
while ((2.15) is **false**) **and** $(i < T)$ **do**
  classify $\{(y_a^{t+1})^{i+1}\}$ based on $(\boldsymbol{\theta}^{t+1})^i$ (2.11);
  estimate $(\boldsymbol{z})^{i+1}$ (2.12);
  localize $\boldsymbol{\alpha}_{(\boldsymbol{z})^{i+1}}$ and obtain $\{p(y|\boldsymbol{\alpha}^t_{(\boldsymbol{z})^{i+1}})\}$
  accept $\{(y^{t+1})^i\}$ based on DM (2.13)(2.14);
  retain Random Forest $(\boldsymbol{\theta}^{t+1})^{i+1}$ (2.10);
  $i = i + 1$;
end while
update $\boldsymbol{\alpha}$ (2.16);
update $\boldsymbol{D}^t$;

(2.5), label $(y_s^{t+1})^{i+1}$ from the structure side will converge to the one with the highest probability, which results in a conflict. Thus, an alternative method using a Decision Maker (DM) is proposed below.

Decision Maker

The labels are accepted using two criteria:

First, the DM only accepts class $k \in \mathbb{Y}$ whose sum of probabilities decreases compared with historical data as

$$\sum_{\left(\boldsymbol{D}^{t+1}_{\boldsymbol{z}_{(+/-)}}\right)^{i+1}_k} p\left(\left(y_a^{t+1}\right)^{i+1}|\boldsymbol{\alpha}^{\boldsymbol{t}}_{(\boldsymbol{z})^{i+1}}\right) < \sum_{(\boldsymbol{D}^t)_k} p\left(y^t|\boldsymbol{\alpha}_{y^t}\right) \tag{2.13}$$

where $(\boldsymbol{D}^t)_k = \{y \in \boldsymbol{D}^t | y = k\}$ and $(\boldsymbol{D}^{t+1}_{\boldsymbol{z}_{(+/-)}})^{i+1}_k$ denotes the current patches whose labels equal to k and they are covered by the foreground or background grid. The difference between $(\boldsymbol{D}^{t+1}_{\boldsymbol{z}_{(+/-)}})^{i+1}_k$ and $(\boldsymbol{D}^{t+1})^{i+1}_k$ is that the former denotes the patches classified into class k and covered by a corresponding grid, like the foreground $(\boldsymbol{D}^{t+1}_{\boldsymbol{z}_+})^{i+1}_k$, while the later one illustrates all patches with label k in frame $t+1$ and iteration $i+1$. Here, $\boldsymbol{\alpha}_{y^t}$ denotes historical parameter when data item $(\boldsymbol{x}^t, y^t)$ is determined to be updated into $\boldsymbol{D}^t$ and $p(y^t|\boldsymbol{\alpha}_{y^t})$ is this data item's probability.

This criterion is based on the assumption that a class of patches should represent a semantic appearance region of a target as shown in Figure 2.4(c)-(d), which would enforce the sum of probabilities to be consistent throughout the sequence. Particularly, the classification boundary of class k will tend to expand during training if additional new data items are included into this class' training set. To prevent unlimited expansion of the boundary in the whole iteration, only the classes with smaller number of data items are chosen. Since the appearance between successive frames is similar, the decrease of the sum of probabilities indicates that some patches that should occupy on particular regions disappear from the observation.

Second, in the chosen class k from the first step, a certain percentage $(\pi_k)^{i+1}$ of data items are accepted starting from the highest probability, which is calculated as follows:

$$\begin{aligned}(\pi_k)^{i+1} = & \left[1 - \frac{\sum_{\left(\boldsymbol{D}^{t+1}_{\boldsymbol{z}_{(+/-)}}\right)^{i+1}_k} p\left((y_a^{t+1})^{i+1} | \boldsymbol{\alpha}^t_{(\boldsymbol{z})^{i+1}}\right)}{\sum_{(\boldsymbol{D}^t)_k} p(y^t|\boldsymbol{\alpha}_{y^t})}\right] \\ & * \min\left[1, \frac{\sum_{\left(\boldsymbol{D}^{t+1}_{\boldsymbol{z}_{(+/-)}}\right)^{i+1}} p\left((y_a^{t+1})^{i+1} | \boldsymbol{\alpha}^t_{(\boldsymbol{z})^{i+1}}\right)}{\sum_{(\boldsymbol{D}^t)} p(y^t|\boldsymbol{\alpha}_{y^t})}\right]\end{aligned} \tag{2.14}$$

The first term of (2.14) denotes the missing percentage of k compared to historical data $(\boldsymbol{D}^t)_k$. The second term compares the sum of probabilities of two grids with historical data, providing a weight for the percentage of the first term. For the first term, the lower the probability of the class, the higher the percentage of acceptance of the class is. For the second term, if every class is observed less, the weight will be smaller.

The goal of collaboratively minimizing both appearance and structure losses is to search for an appropriate classification boundary architecture in feature space that fits the inner similarity between historical patches with current ones and makes the classification results match the inner structure of the object simultaneously. Thus, adding some new and credible observed patches into training set becomes necessary. The criterion of (2.14) chooses only the data items associated with high probabilities. The intermediate result of accepted patches are shown in Figure 2.4 (e)–(f).

To terminate the optimization, DM checks if there is any improvement in performance as

$$\begin{cases} \sum\limits_{\left(\boldsymbol{D}_{\boldsymbol{z}_+}^{t+1}\right)^{i+1}} p\left(\left(y_a^{t+1}\right)^{i+1}|\boldsymbol{\alpha}_{(\boldsymbol{z})^{i+1}}^t\right) < \sum\limits_{\left(\boldsymbol{D}_{\boldsymbol{z}_+}^{t+1}\right)^{i}} p\left(\left(y_a^{t+1}\right)^{i}|\boldsymbol{\alpha}_{(\boldsymbol{z})^{i}}^t\right) \\ \sum\limits_{\left(\boldsymbol{D}_{\boldsymbol{z}_-}^{t+1}\right)^{i+1}} p\left(\left(y_a^{t+1}\right)^{i+1}|\boldsymbol{\alpha}_{(\boldsymbol{z})^{i+1}}^t\right) < \sum\limits_{\left(\boldsymbol{D}_{\boldsymbol{z}_-}^{t+1}\right)^{i}} p\left(\left(y_a^{t+1}\right)^{i}|\boldsymbol{\alpha}_{(\boldsymbol{z})^{i}}^t\right) \end{cases} \tag{2.15}$$

As discussed in Section 2.3.3, an alternative way to estimate the global performance is to make use of the sum of probabilities for both foreground and background, since the classification quality is impossible to measure without true labels. If there is no increase in the sum in neither foreground nor background, it is considered as unimproved and the optimization is terminated. Besides, this optimization is also stopped if a maximum iteration number T is reached. The performance of this optimization

process is discussed in Section 2.5.2 and illustrated in Figure 2.9 and 2.8.

2.3.5 Update

In this structured prior regularized semi-supervised learning framework, the label of each patch y^{t+1}, classifier model $\boldsymbol{\theta}^{t+1}$, and target location $\boldsymbol{z}$ can be simultaneously estimated via optimization. If there are still unaccepted patches after optimization, they are assigned the labels with the highest probabilities of structure model by (2.5). The only parameter that still needs to be explicitly updated is the α_k^{t+1} value of the distribution. If the estimated label of a vertex in optimization is k, the update is given by (2.16). Here, $\boldsymbol{\alpha}^{t+1}$ remains the same as $\boldsymbol{\alpha}^t$ for the vertex whose patch is not accepted in the optimization. That is,

$$\alpha_k^{t+1} = \alpha_k^t + 1 \tag{2.16}$$

To update the new data items into historical data, the system stores a fixed size data item pool for each class. The patches $\{\boldsymbol{x}^{t+1}\}$, their labels $\{y^{t+1}\}$ and the corresponding probabilities generated by the structure model $\{p(y^{t+1}|\boldsymbol{\alpha}_{y^{t+1}})\}$ are updated into these pools. After one frame is tracked, if a new patch accepted in the optimization has a probability higher than the ones in the pool, this new data item replaces the one with the lowest probability. In other words, the pool always keeps the set of data items with the highest probabilities. Through this mechanism, the proposed algorithm provides a reliable measure for updating.

2.4 Implementation Details

To handle the target's varying size, an inert size update strategy is applied in the system based on three factors, namely, the classified labels in the final iteration of the optimization, the final estimated target position, and $-\boldsymbol{L_s}$ in (2.3) which measures the consistency between appearance and structure. To avoid increasing uncertainty, the target size will be updated only when the value of $-\boldsymbol{L_s}$ increases compared with the last frame and 80% foreground vertices are classified as foreground. If the above condition is satisfied, labels around the four borders of the foreground grid are examined. If all labels of the vertices on the borders are classified as background, the foreground grid removes one vertex from each direction. Conversely, if all labels of background vertices adjoining the borders are classified as foreground, the grid adds one vertex.

The Random Forest is built by 50 independent decision trees. Its information gain [24] for training the split function is defined as,

$$ig(\boldsymbol{D}) = H(\boldsymbol{D}) - \frac{|^l\boldsymbol{D}|}{|\boldsymbol{D}|}H(^l\boldsymbol{D}) - \frac{|^r\boldsymbol{D}|}{|\boldsymbol{D}|}H(^r\boldsymbol{D}) \tag{2.17}$$

where $H(\boldsymbol{D})$ is the entropy of the data set reaching the internal node and $|\boldsymbol{D}|$ denotes the number of those data items in $\boldsymbol{D}$. In our implementation, two terminal criteria are used. First, the maximum depth of each tree is set to 20. Second, if only the features of one class are left in a node, the growing process of this branch is terminated. Two kinds of split functions [24] are randomly chosen in training, namely decision dump and two-dimension linear classifier. Every tree is trained with a complete training set.

2.4.1 Adaptive Extension

The main computational cost of this proposed algorithm comes from the repeated training of Random Forest. To speed up, an adaptive Random Forest is implemented.

Suppose there is a different data set $(\boldsymbol{D})^{i+1}$ compared with the set $(\boldsymbol{D})^{i}$ in the last training iteration i. The idea is to make use of the difference between the two sets to partially re-train each tree. The key is to determine which sub-tree identified by a node needs to be pruned and re-grown. This procedure for a tree is described as follows: 1) All nodes whose data sets changed are marked and their information gains for all non-leaf-nodes and entropy for the leaves are re-calculated. 2) All marked nodes are assigned weights $\{\omega\}$, in which the weight of the leaf with maximum depth is set to 0. 3) A sampling mechanism is developed to choose the node to re-train its corresponding sub-tree. Each time, only one node is sampled according to the weights. The identities of the marked nodes belonging to the re-trained sub-tree and containing the sub-tree are reset. 4) Go back to step 3) until there is no more marked node.

The weight for a non-leaf-node is defined as

$$\omega = \max\left[0, ig\left((\boldsymbol{D})^{i+1}\right) - ig\left((\boldsymbol{D})^{i}\right)\right] \tag{2.18}$$

The insight underling this definition is that, if previous information gain is larger than the new one, the condition of this node is getting worse, which should consequently have a higher chance to be re-trained. Closer to the root node, the influence by new data set tends to be smaller, which makes it harder for the algorithm to modify upper internal nodes. The weight for a leaf whose depth does not reach the maximum is

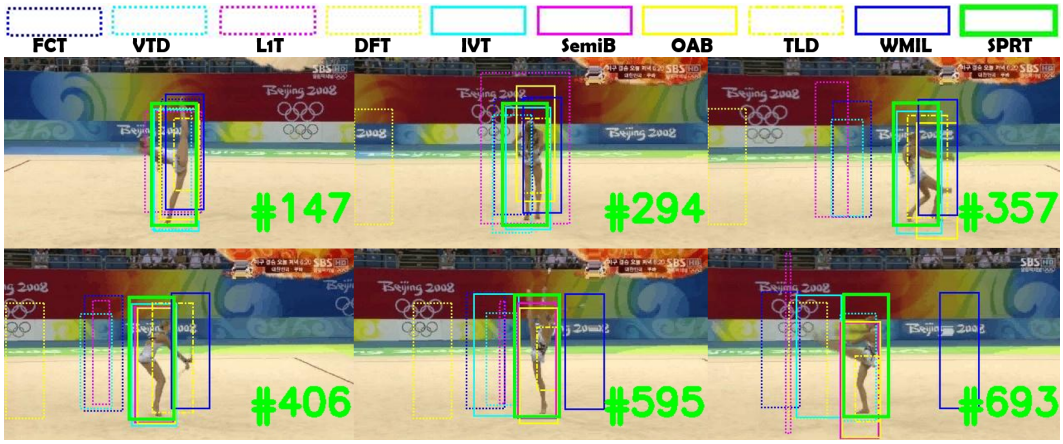

Figure 2.5: Sample tracking results for sequence *gym*

defined as

$$\omega = \max\left[0, H\left((\boldsymbol{D})^{i+1}\right) - H\left((\boldsymbol{D})^{i}\right)\right] \tag{2.19}$$

The above equation gives the leaf more chance to continuously grow down.

2.5 Experiments

Our experiment was carried out on more than 60 video sequences that have been previously used in the literature, including 50 videos of the visual tracking benchmark in [128] and another 12 video clips from [56][127], whose names are given in Figure 2.10. Besides the benchmark trackers, the author have also conducted experiments by comparing with three other recently proposed state-of-the-art trackers namely, least soft-threshold squares tracker (LSST) [117], understanding and diagnosing visual tracking system (UDT) [120] and KCF [47], to demonstrate the effectiveness of the proposed approach. Note that only the results of top 10 trackers are shown. On the extra 12 videos, nine methods were compared, namely, visual tracking decomposition

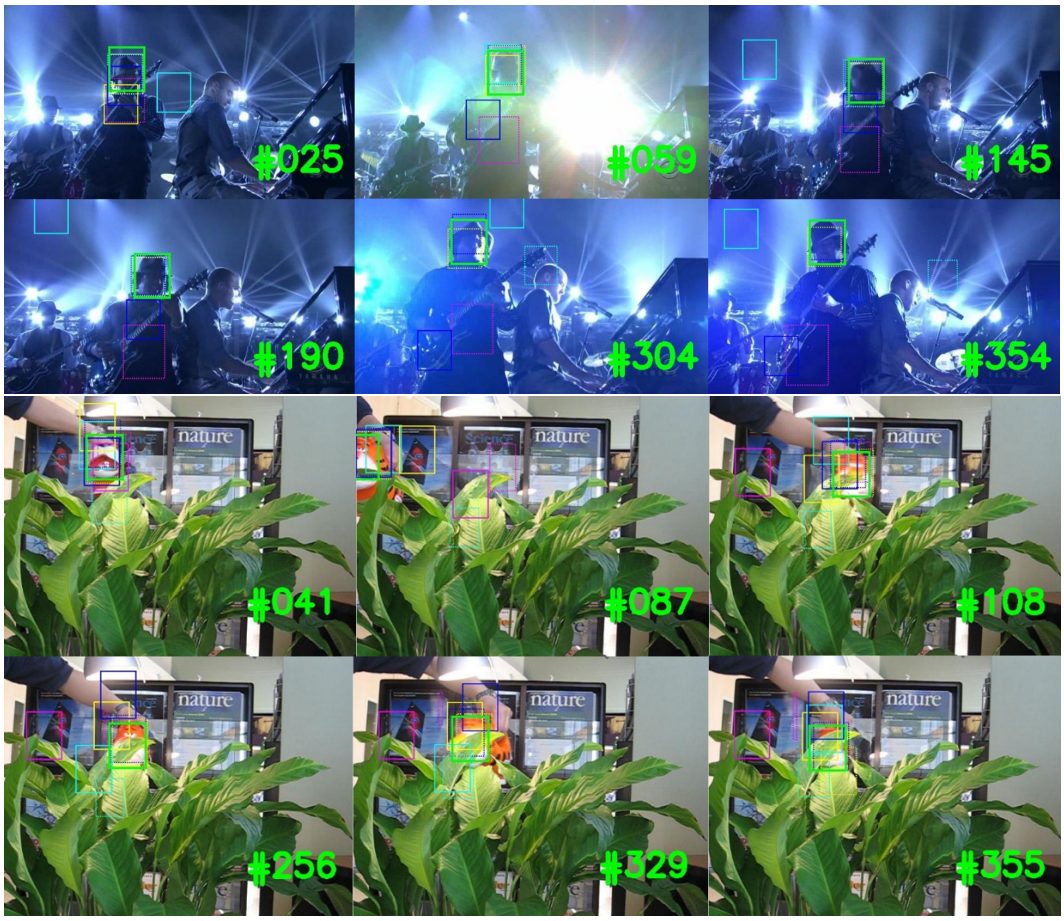

Figure 2.6: Sample tracking results for sequences *shaking* and *tiger2*

algorithm (VTD) [55], incremental visual tracker (IVT) [94], Semi-supervised tracker (SemiB) [40], weighted MIL tracker (WMIL) [134], the online AdaBoost method (OAB) [39], TLD tracker (TLD) [53], distribution field tracker (DFT) [96], l_1 tracker (L1T) [81] and the fast compressive tracker (FCT) [135].

The feature vector of each 16×16 patch is composed of two parts, one directly stretching the patch into a vector of 768 bins and the other extracting one block of HOG feature with a 2×2 block and 8×8 cell. The algorithm version without HOG feature denoted as SPRT(C) is also used for comparation. Foreground patches

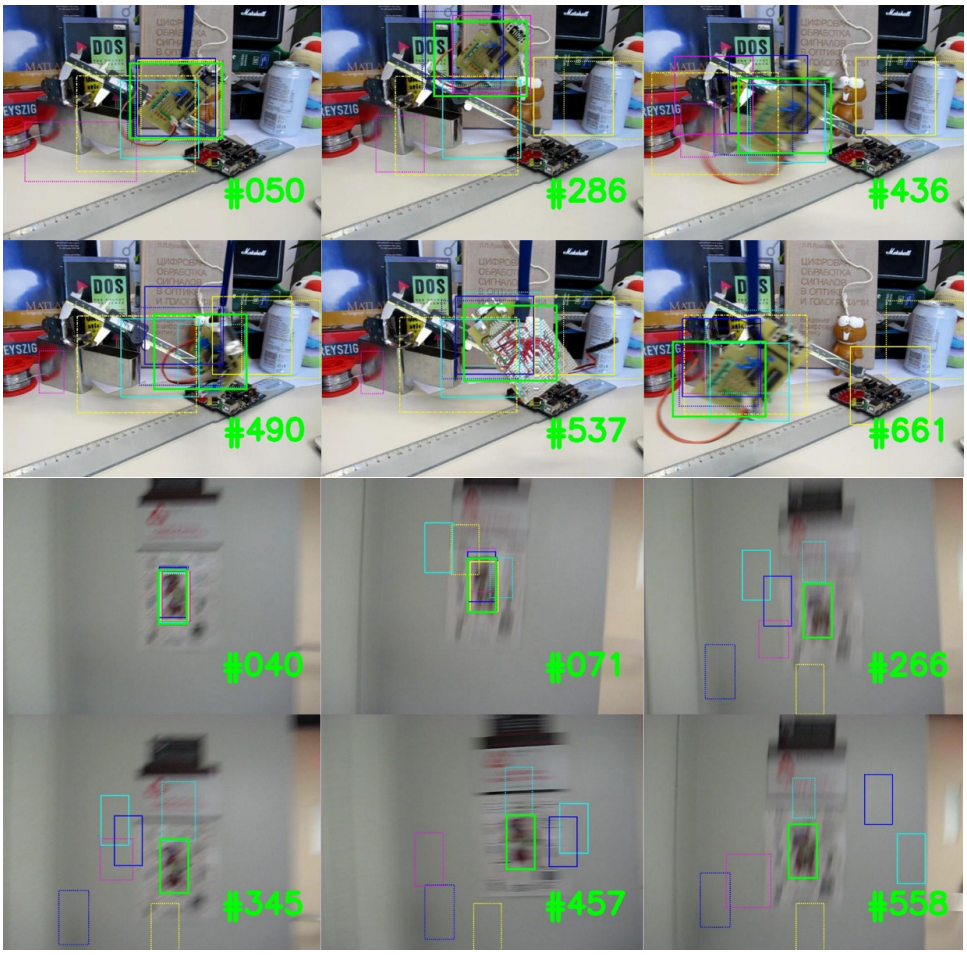

Figure 2.7: Sample tracking results for sequences *board* and *tu-owl*

are clustered into 7 classes and background into 10 classes. The distance between adjacent vertices on the grid is set to 3 pixels. The Dirichlet-multinomial distribution parameter is uniformly initialized as $\mathbf{1}$, a 17×1 vector. In searching $\boldsymbol{z}$ process, 500 candidate positions are sampled from a normal distribution with covariance $[9, 0; 0, 9]$ based on the pre-estimated location. The only input is the bounding box in the first frame. All 17 classes for the foreground and the background are automatically clustered using the k-means method on the first frame. The maximum iteration T is set to 20.

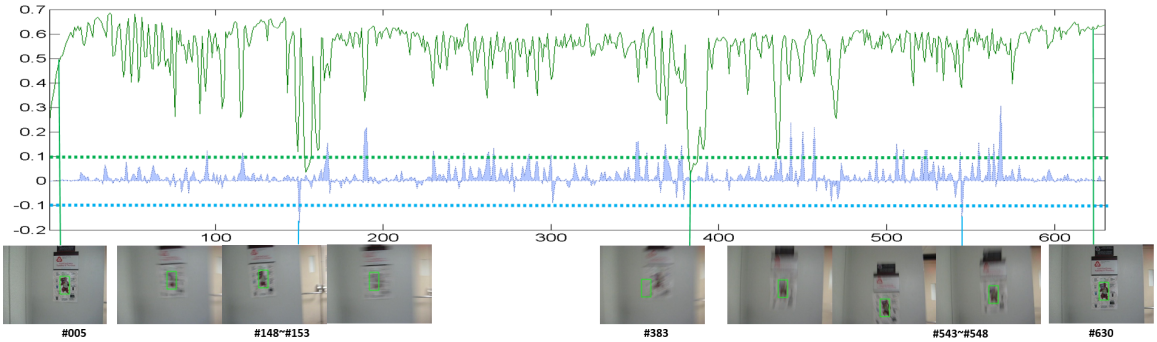

Figure 2.8: Optimization performance

The algorithm was implemented in MATLAB. Experiment runs on an Core (TM) i7 CPU running at 3.5 GHz with 12 GB memory. Currently, this implementation is not optimized and the major computational cost comes from optimization. The computational time depends on the number of trees and the iteration times. According to randomly chosen 50 frames, the proposed algorithm takes 3.1 seconds on average per iteration. In more than 90% of the frames, the tracking procedure needs 1 to 4 iterations and less than 10% needs 5 to 20 iterations. The proposed adaptive Random Forest saves 38.5% CPU time for a single tree on average when compared with the standard version.

2.5.1 Evaluation Methodology

Two metrics are used to evaluate the proposed algorithm. The first one is the success rate, which is used in the PASCAL VOC [35] challenge and defined as score= $\frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)}$ where ROI_T is the tracking bounding box and ROI_G is the ground truth bounding box. For comparison, the author count the number of frames whose overlap score is larger than the given threshold, i.e., they are deemed successful. The success plot shows the ratios of successful frames as the threshold varies from 0

to 1. To rank the trackers in the two datasets, the threshold is fixed at 0.5. The other one is the center location error, which is defined as the Euclidean distance between the central location of the tracked objects and the manually labelled ground truth. The precision plot is adopted to measure the tracking performance in the benchmark. It shows the percentage of frames whose estimated location is within the given threshold distance from the ground truth. The threshold for ranking is set at 20 pixels. For the extra dataset, the average center location error over all frames of one sequence is utilized. All parameters in the proposed algorithm are fixed for all experiments to demonstrate the robustness and the stability. For the other trackers, the source or binary codes provided by the authors with corresponding default parameters were used. All trackers are initialized with the same initial parameters.

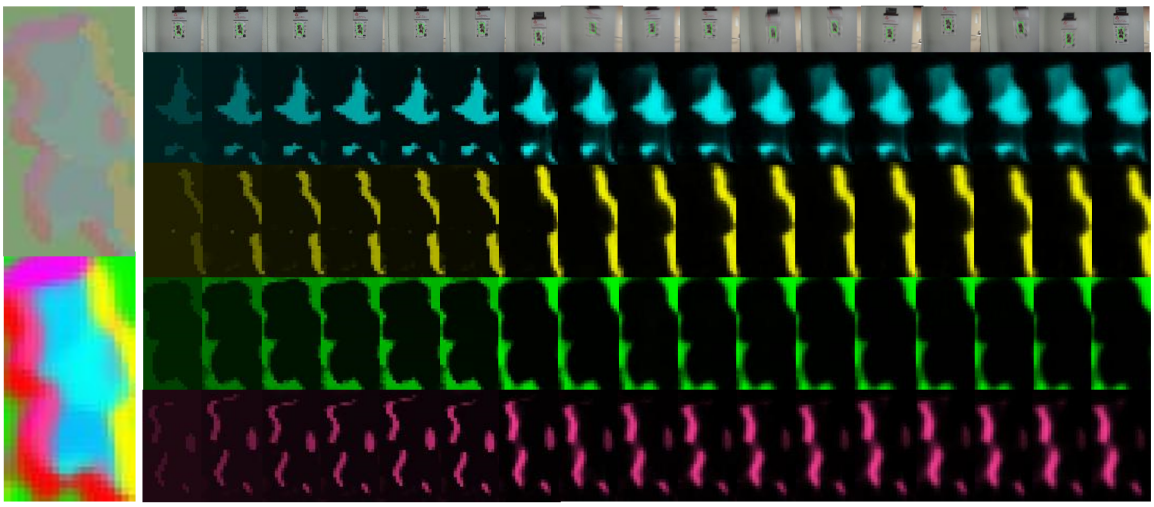

Figure 2.9: Evolution of the distributions: on the left side, Frame #0 and Frame #630; on the right side, frame numbers of *tu-owl* are #0, #3, #6, #9, #12, #15, #65, #115, #165, #215, #265, #315, #365, #415, #465, #565, #630; the complete illustration can be found in Section 2.5.2.

2.5.2 Model Analysis

Effect of Optimization: Using Figure 2.8, the intuition behind the optimization process can be illustrated. The green solid curve representing $-\boldsymbol{L_s}$ in (2.3) indicates how well the foreground grid explains the given classification results. The blue dashed curve (with blue area) shows the difference in $-\boldsymbol{L_s}$ between the last and the first iterations of optimization. The optimization is tested in this manner because the classification performance $\boldsymbol{L_a}$ is impossible to measure independently without the true labels, which is also why the algorithm cannot employ traditional gradient-based method for multi-objective optimization.

It can be observed that the structure model expresses the target well at the beginning and the green solid curve tends to oscillate as the target becomes blurred. If this value is above 0.1, it means correct tracking as in Frame #005 and Frame #630. If it is lower than 0.1, it may mean incorrect tracking as in Frame #383. During oscillation, tracking performance degrades at first and then recovers with values between 0.4 and 0.7. This is because the classifier adapts to the blurred appearance after several updates.

As shown in the blue dashed curve, there are 467 positive values and 163 negative ones. The negative values indicate that the algorithm attempts to optimize the appearance part while sacrificing the structure model. This happens when a clear target appears during a blurred period as in Frame #148 to Frame #153 and Frame #543 to Frame #548. It can also be observed that when the green solid curve is stable, the blue one displays the same behavior, since there is little room to improve. The difference caused by exploiting the current frame remains between -0.1 and 0.1. The reason is that the algorithm attempts to optimize two conflicting objectives

Seq.	SPRT	SPRT(C)	OAB	SemiB	IVT	WMIL	TLD	DFT	L1T	VTD	FCT
tu-face	100/7.9	100/6.7	31.0/83.9	75.2/99.3	23.7 /112	20.8/127	100/3.9	29.0/75.3	16.2/99.2	100/5.3	25.9/74.3
biker	99.3/4.31	99.1/4.33	66.0/10.0	39.0/14.0	10.0/111	6.84/38.8	2.00/166	9.4/78.1	4.27/83.2	15.0/86.0	85.0/6.00
gym	94.8/9.5	96.7/8.78	82.9/13.4	57.7/107	64.2/21.1	17.3/51.6	20.7/102	22.4/39.2	22.4/39.2	36.9/30.2	20.8/46.4
tu-car4	100/5.22	100/5.32	96.5/27.1	72.1/113	58.4/80.6	5/177	12.6/88.2	100/5.94	5.5/227	100/3.2	58.4/80.6
tu-car3	100/3.29	100/8.14	40.3/221	82.6/65	32.2/171	28.0/43.2	79.2/29.0	10.3/136	23.5/237	50.7/88.7	32.4/177
torus	95.5/5.2	93.18/6.5	4.9/110	7.9/129	20.8/39.7	25.3/47.4	7.5/60.8	56.8/32.3	15.5/52.3	93.5/4.2	10.2/67.5
tu-car2	92.9/6.2	91.7/5.63	13.3/155	84.7/49.3	26.1/152	14.7/132	86.6/20.2	12.9/255	13.1/207	74.5/51.5	13.3/258
tu-owl	91.7/9.87	96.6/5.57	15.6/355	33.2/281	7.13/137	16.4/95.9	62.2/40.3	9.3/172	9.3/116	8.7/98.2	20.4/175
board	88.9/14.3	75.2/34.3	21.1/113	19.3/354	20.1/116	9.7/67.4	10.7/156	24.7/141	3.87/276	62.7/42.8	7.7/92.1
box	89.3/25.9	88.6/25.9	20.7/92.5	31.6/265	26.2/135	27.9/69.7	78.2/22.7	24.0/120	9.21/82.2	39.5/110	24.8/114
tu-car1	95.5/7.9	99.8/3.9	46.2/225	63.0/153	20.4/164	1.2/151	0.1/99.8	7.8/84.2	46.2/119	1.3/236	1.0/155
dr-baby	83.2/20.1	84.0/10.2	17.7/88.8	6.1/116	33.6/55.2	5.3/68.4	3.5/102	11.5/76.9	24.7/75.7	57.5/24.6	22.1/70.7

Figure 2.10: Success rate / average center location error

simultaneously.

Evolution of Distributions: Using Figure 2.9, the evolution of the set of structured Dirichlet-multinomial distributions is examined to demonstrate the capability to update the positive and negative data items. In the figure, one color represents one class/label occupying some vertices. The colors in the image are calculated by multiplying the original RGB values by the probabilities of the corresponding labels. Higher probability makes the pixel lighter and vice versa.

On the left side, seven labels are illustrated via linear combination. The top one comes from the first frame and the bottom one from the last frame. On the right side, the evolutions of the four labels are shown by sampling 17 frames. It shows that 1) although each vertex is independently treated in the structure model, each class represents its own semantic region well in the absence of any dramatic change in the whole sequence; 2) although each distribution is initialized as uniform, the algorithm makes it concentrate on one particular label. Since the set of structured distributions is utilized to regularize the optimization for acquiring new classification model and it provides a measure to update features, the system itself maintains consistency in tracking without being sensitive to noisy estimates.

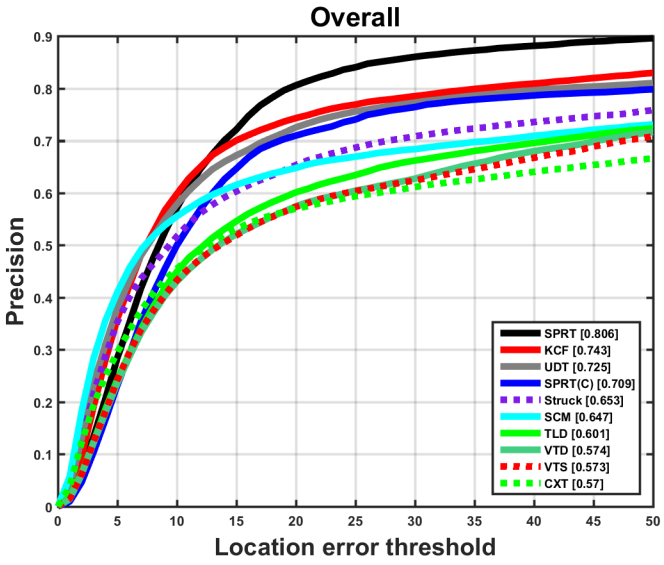

Figure 2.11: Plots of center location error for the complete dataset of benchmark [128]

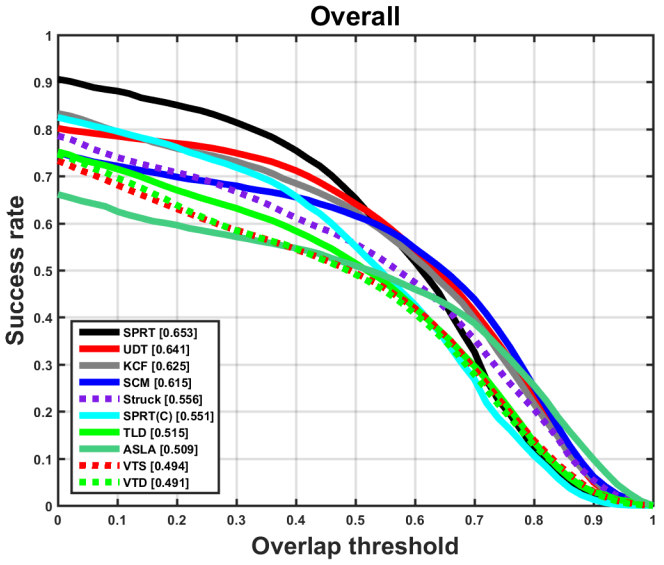

Figure 2.12: Plots of success rate for the complete dataset of benchmark [128]

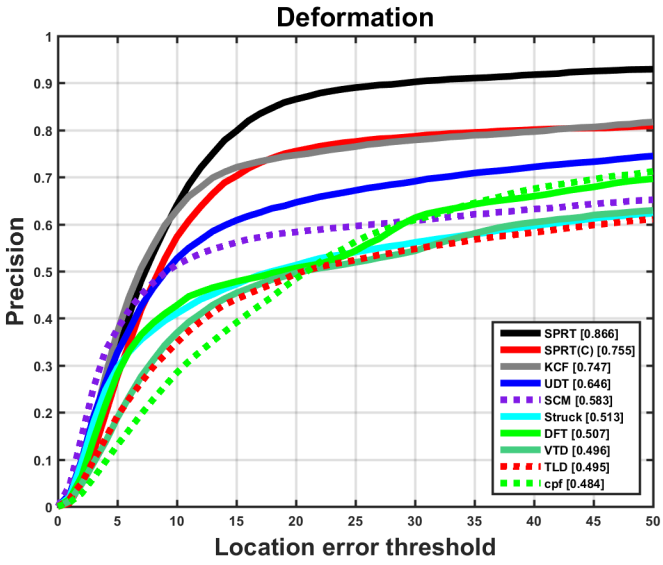

Figure 2.13: Plots of center location error for the challenge *deformation* of benchmark [128]

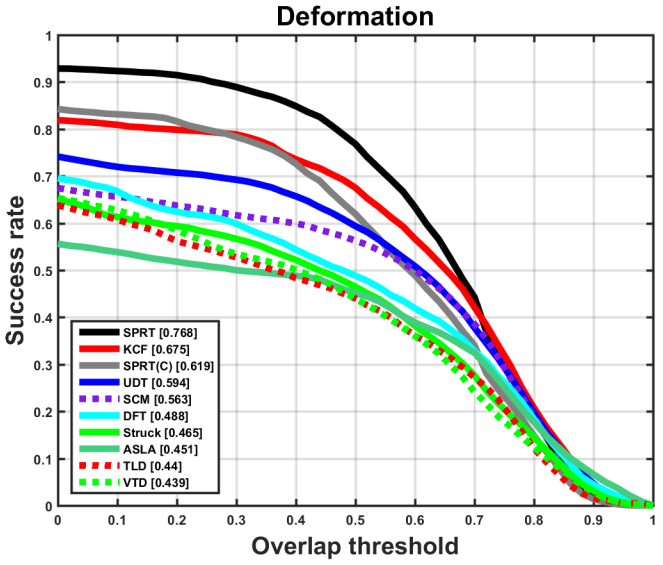

Figure 2.14: Plots of success rate for the challenge *deformation* of benchmark [128]

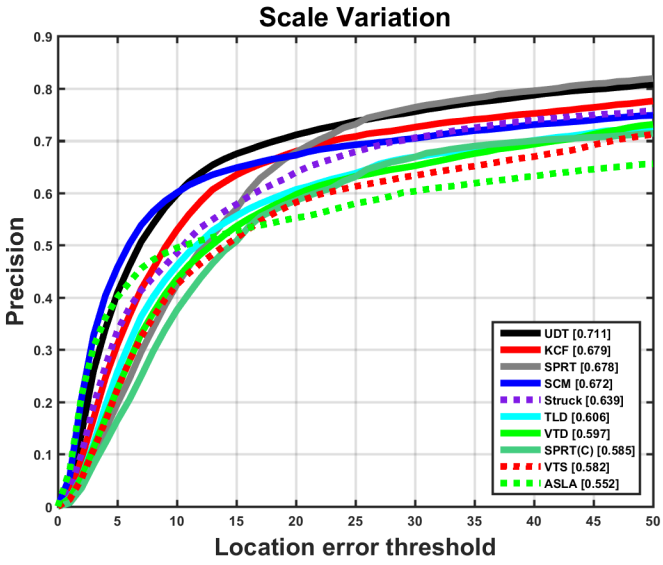

Figure 2.15: Plots of center location error for the challenge *scale variation* of benchmark [128]

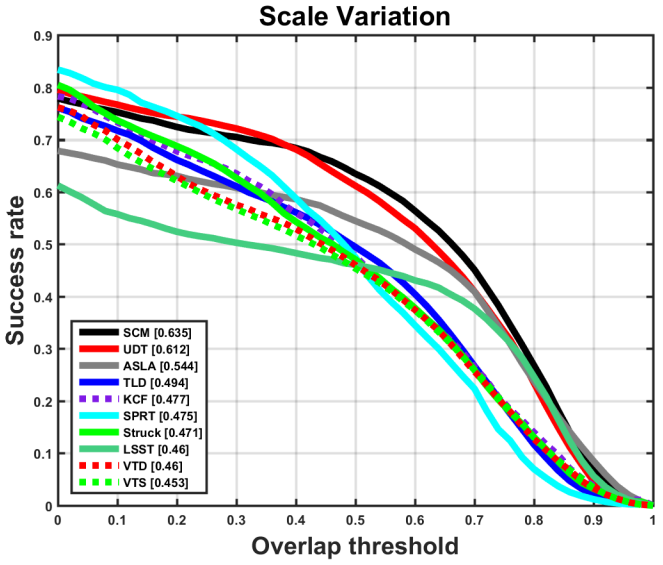

Figure 2.16: Plots of success rate for the challenge *scale variation* of benchmark [128]

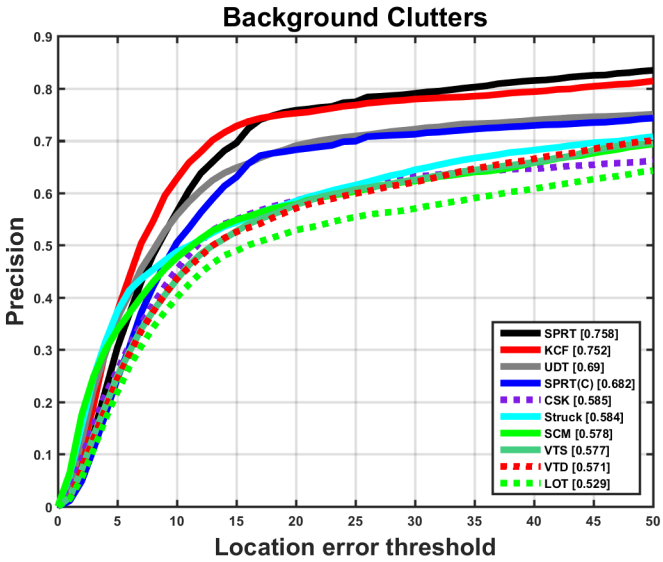

Figure 2.17: Plots of center location error for the challenge *background clutters* of benchmark [128]

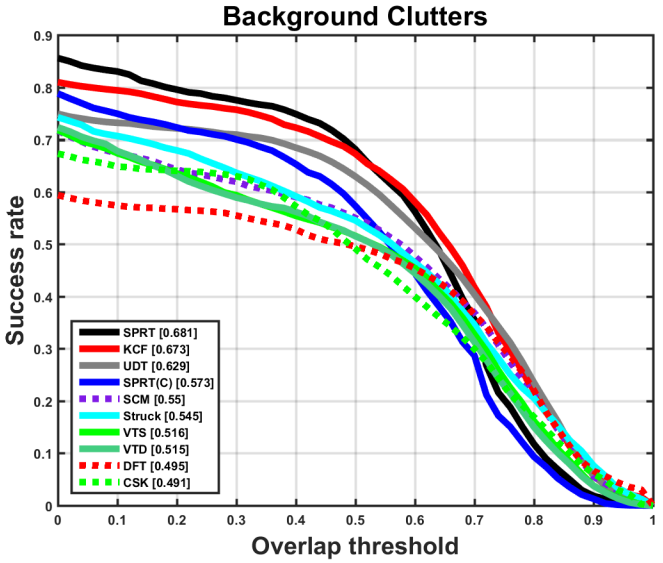

Figure 2.18: Plots of success rate for the challenge *background clutters* of benchmark [128]

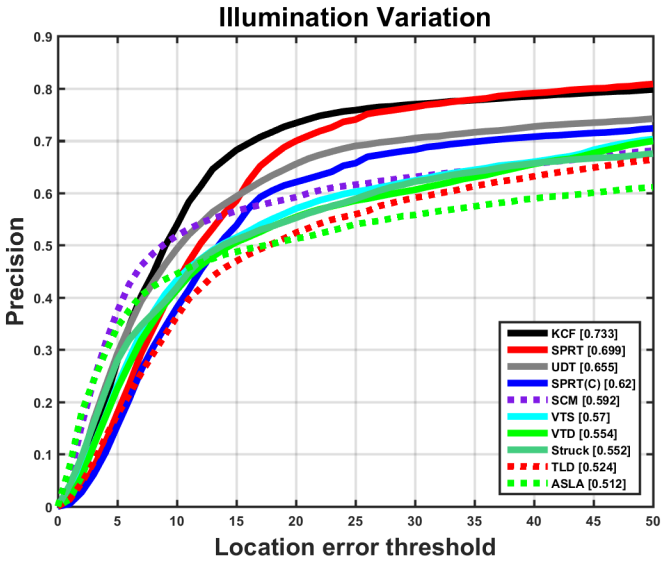

Figure 2.19: Plots of center location error for the challenge *illumination variation* of benchmark [128]

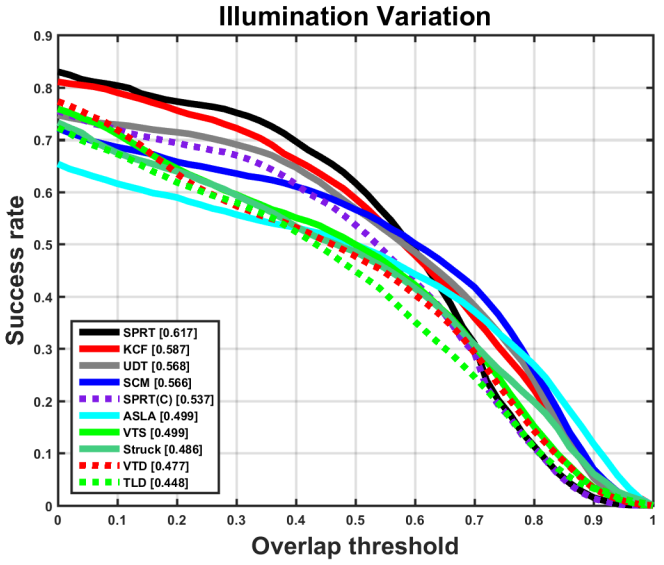

Figure 2.20: Plots of success rate for the challenge *illumination variation* of benchmark [128]

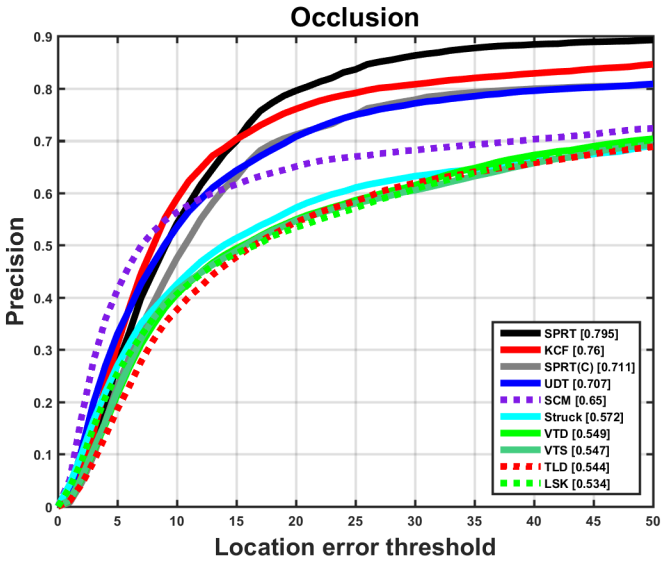

Figure 2.21: Plots of center location error for the challenge *occlusion* of benchmark [128]

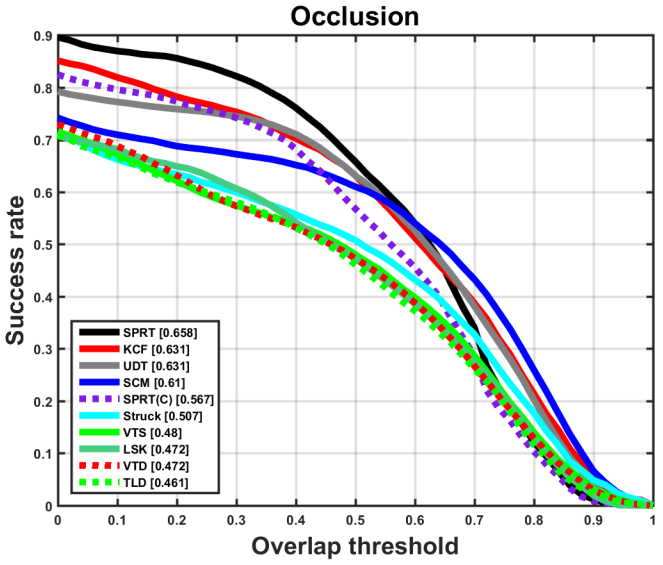

Figure 2.22: Plots of success rate for the challenge *occlusion* of benchmark [128]

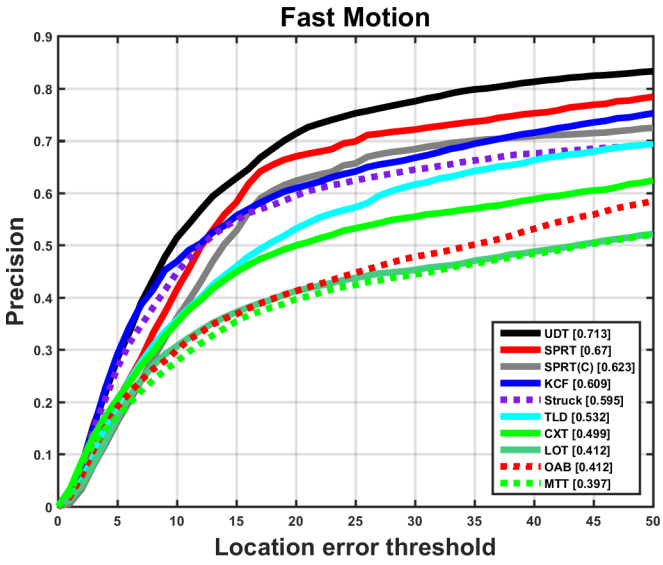

Figure 2.23: Plots of center location error for the challenge *fast motion* of benchmark [128]

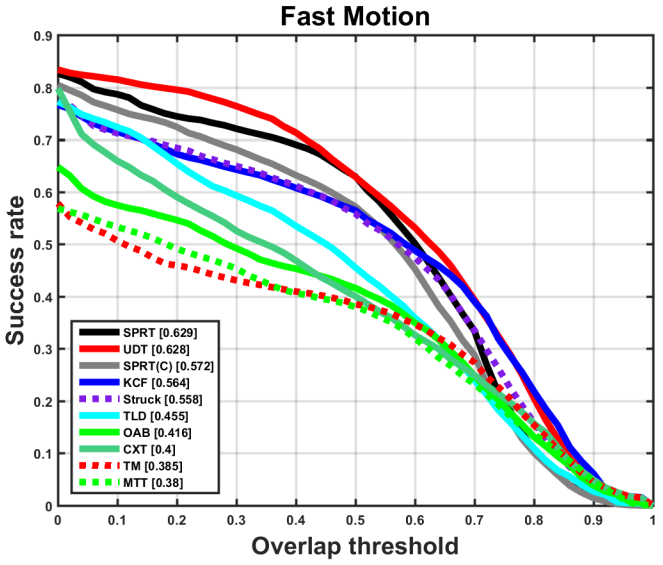

Figure 2.24: Plots of success rate for the challenge *fast motion* of benchmark [128]

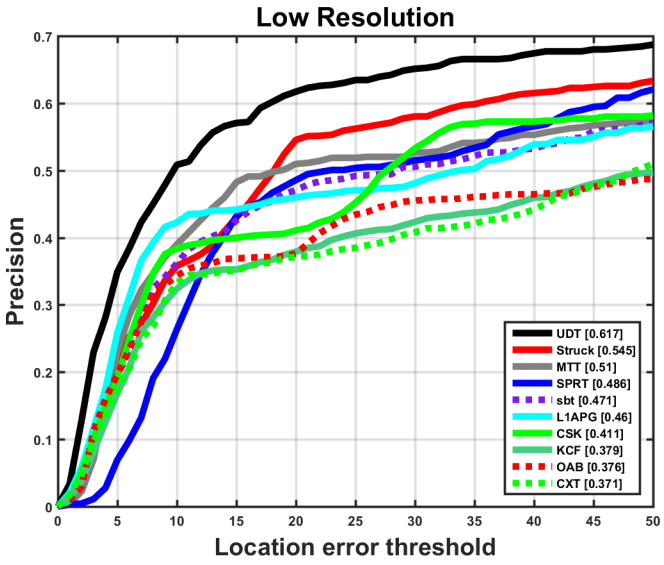

Figure 2.25: Plots of center location error for the challenge *low resolution* of benchmark [128]

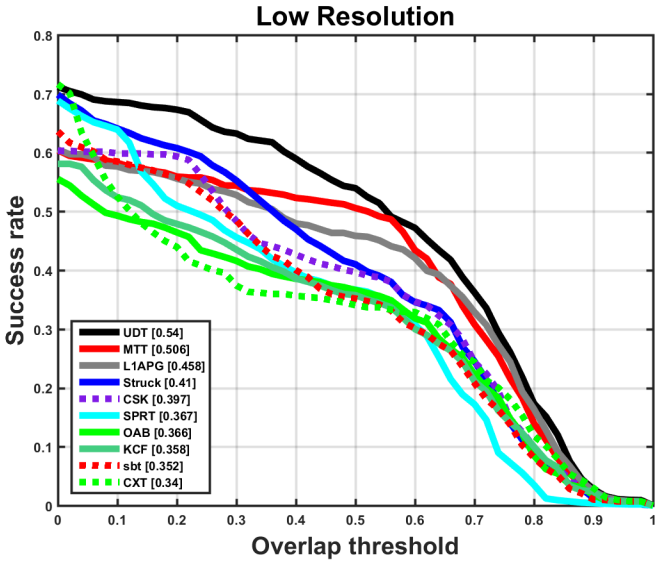

Figure 2.26: Plots of success rate for the challenge *low resolution* of benchmark [128]

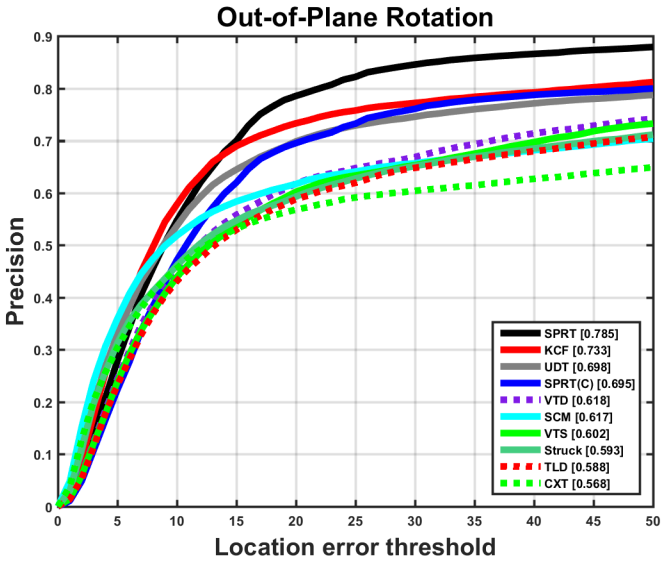

Figure 2.27: Plots of center location error for the challenge *out-of-plane rotation* of benchmark [128]

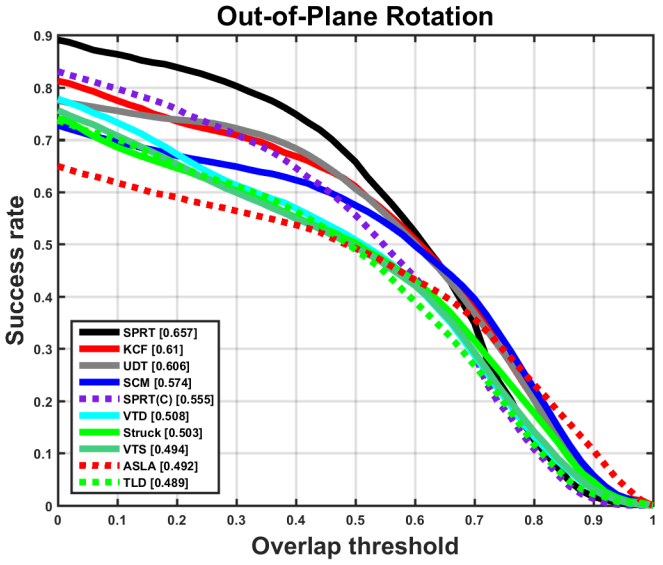

Figure 2.28: Plots of success rate for the challenge *out-of-plane rotation* of benchmark [128]

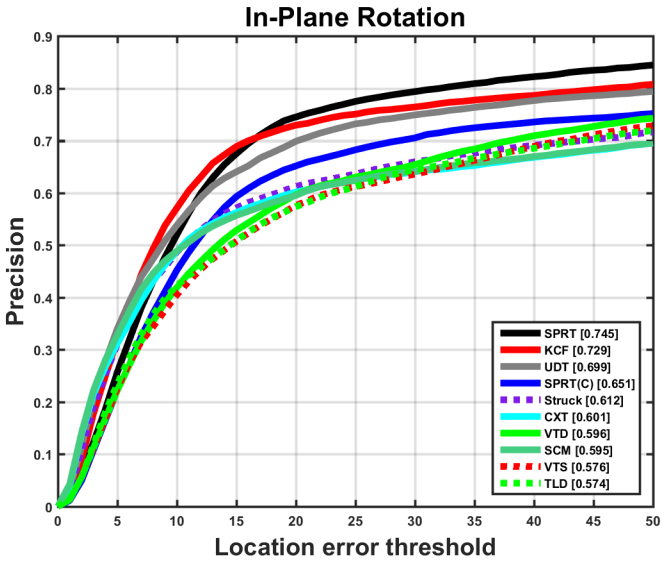

Figure 2.29: Plots of center location error for the challenge *in-plane rotation* of benchmark [128]

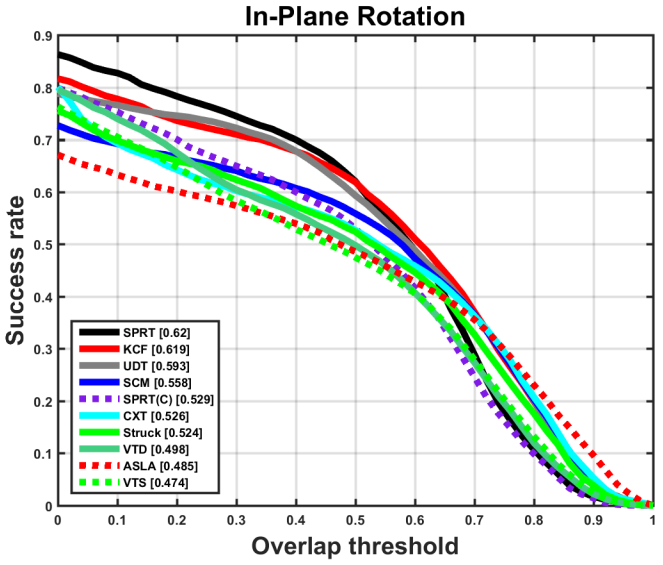

Figure 2.30: Plots of success rate for the challenge *in-plane rotation* of benchmark [128]

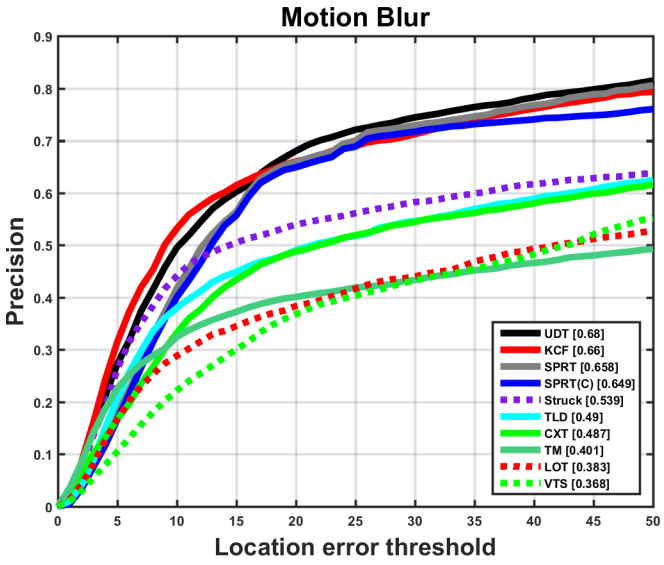

Figure 2.31: Plots of center location error for the challenge *motion blur* of benchmark [128]

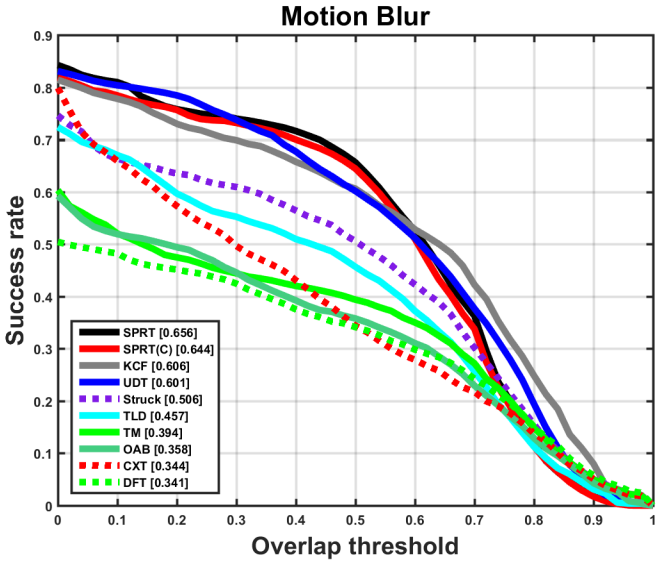

Figure 2.32: Plots of success rate for the challenge *motion blur* of benchmark [128]

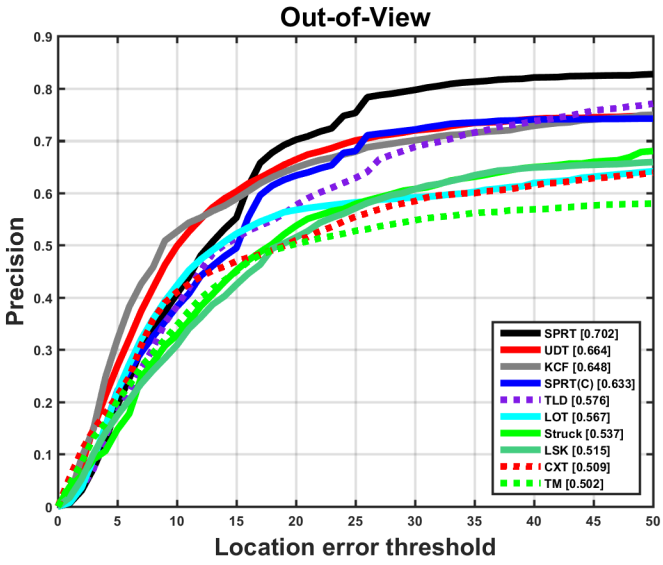

Figure 2.33: Plots of center location error for the challenge *out-of-view* of benchmark [128]

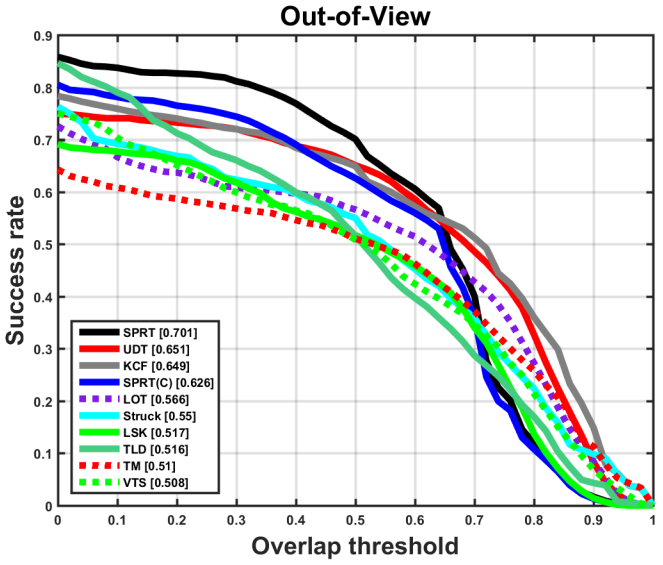

Figure 2.34: Plots of success rate for the challenge *out-of-view* of benchmark [128]

2.5.3 Performance Evaluation

Benchmark tracking results are shown in Figure 2.11 to Figure 2.34. In both the overall precision plot and the overall success plot, the proposed method SPRT ranks the first. The SPRT(C) also outperforms most methods of the benchmark. Moreover, 12 additional video clips are utilized to evaluate SPRT's performance. The results are shown in Figure 2.10. The proposed algorithm achieves the highest success rate (red) in 8 videos and the lowest average center location error (red) in 3 videos. In most of video clips where SPRT is not ranked the best, it is the second best (blue).

Some results from the proposed method and those from alternative methods are shown in Figure 2.5, Figure 2.6 and Figure 2.7. In *gym*, for example, the pose of the gymnast varies as she rotates, which results in the target undergoing deformation as in Frame #357. In the *shaking* sequence, the scenario suffers from serious illumination changes. In most frames, the appearance changes gradually due to lighting, while in some frames, the target undergoes severe illumination changes as in Frame #59. In the sequence *tiger2*, the target is partially occluded. In the video clip *board*, the target is moving through a cluttered background. The scenario consists of rich texture, which makes tracking more challenging. In *tu-owl*, blur and abrupt motions occur. Nevertheless, the experiment shows that SPRT handles these situations well, which demonstrates the effectiveness of the proposed prior-regularized semi-supervised learning framework. Moreover, our method shows robustness on the whole *tu-* series sequences, which are especially designed to demonstrate *motion blur*.

In benchmark [128], data sets are annotated with attributes, describing challenges such as *deformation*, *scale variation*, *background clutters*, *illumination variation*, *occlusion*, *fast motion*, *low resolution*, *out-of-plane rotation*, *in-plane rotation*, *motion*

blur and *out-of-view*. To further evaluate the proposed tracker in different situations, all attributes are reported in Figure 2.13 to Figure 2.34. It shows that SPRT outperforms other trackers in more than half the attributes. For example, it ranks the best in both the *occlusion* success plot and the *occlusion* precision plot, which is consistent with the results in Figure 2.5, Figure 2.6 and Figure 2.7. SPRT achieves the best performance in *deformation* attribute. This is because the structured Dirichlet-multinomial distributions make the structure model flexible to random movement of patches and the semi-supervised learning makes the appearance model robust to object appearance variation, which together handle *deformation* well.

The proposed tracker is sensitive to two situations, *scale variation* and *low resolution*, which are shown by the benchmark results. Experiments show that the proposed inert scale update strategy is not effective against ratio change and it tends to keep the current target size. Because of the limited size of the image patch, *low resolution* would highly affect the performance of patch-based trackers. This is because the patch is the observation unit for the whole system and *low resolution* reduces the information that can be extracted from a single patch.

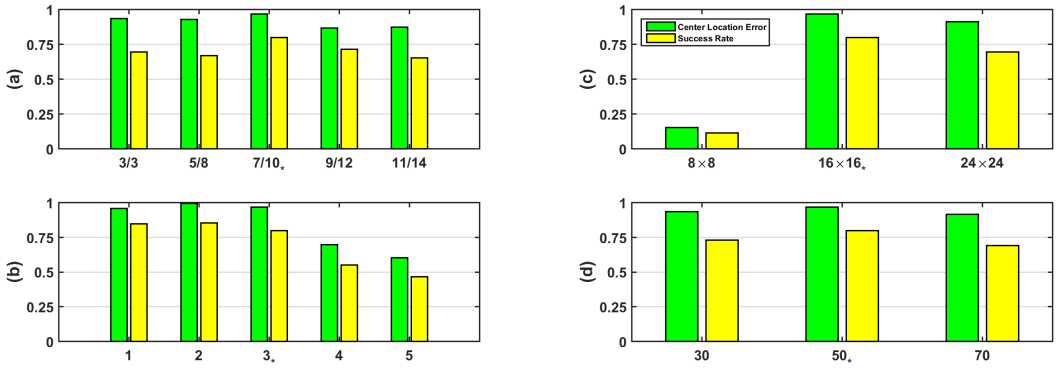

Figure 2.35: Parameter validation: (a) foreground/background class count; (b) distance between vertices; (c) patch size; (d) tree count of Random Forest

2.5.4 Parameter Analysis

Like other tracking methods, SPRT has a number of parameters. The author evaluate the impact of some important parameters, namely, class count or cardinality, density of the grid, patch size and tree count of Random Forest. For this purpose, a set of sequences consisting of *biker*, *tu-car2*, *deer*, *football1* and *couple* is used. In Figure 2.35, the average percentage of frames whose overlap scores are larger than 0.5 (green) and the one with estimated locations within 20 pixels from the ground truth (yellow) are reported. The default parameters are denoted by subscript "$_*$".

The impact of the class count on the performance is depicted in Figure 2.35(a). While all other parameters are fixed, only the class count is changing. This figure demonstrates SPRT's robustness to the variation of class count in the range from 3/6 to 11/14 (foreground/background). Note that the structure model is no longer functional if the class count goes down to one. Figure 2.35(b) shows the effect of the grid density. Despite the jitter in the results, a trend is discernible. The tracking performance starts to deteriorate when the distance between adjunct vertices is larger than 4 pixels. Considering the compromise between speed and performance, the distance 3 is chosen as default. Figure 2.35(c) reveals that the size of the patch significantly affects the algorithm's performance. The tracking system maintains a stable performance with patches no smaller than 16×16. Here, for a 8×8 patch, HOG feature was not extracted. Finally, Figure 2.35(d) demonstrates that the proposed approach works well based on different tree count of Random Forest varying from 30 to 70.

2.6 Conclusion

In this chapter, visual tracking was cast into a prior-regularized semi-supervised learning task. Under the framework, a novel patch-based-grid target representation was designed taking advantage of discriminative appearance model and generative structure model. The appearance model attempted to classify the foreground and the background using not only the historical data but also the current frame data. The set of structured distributions of structure model was formulated as a regularizer to constrain the learning process. A heuristic multi-objective optimization method was proposed to find the solution, which ensures appearance and structure collaboratively optimized.

The experiments on standard sequences showed that the proposed algorithm outperformed existing approaches that especially handle the effects caused by pose, rotation, illumination, blur, abrupt motion, occlusion and background clutter. It was also shown that the use of the current frame could improve tracking performance and that the tracker could adapt itself rather than explicitly updating positive and negative features based on noisy estimate.

Chapter 3

Dynamic Background Subtraction by a Probabilistic Topic Model

Video analysis often starts with background subtraction, aiming to extract any moving objects. This task is usually defined as a binary classification problem based on the assumption that the background dominates major observations while foreground objects tend to be outliers. However, the potential dynamic nature of the background weakens this foundational assumption. In this chapter, the background is formulated as a probabilistic topic model. To make it suitable for application, an innovative on-line topic model variant and an incremental learning method are designed. Using this model, a stable potentially appearing topic set and an adaptive topic proportion are learned in order to represent per-pixel observations, which improves foreground detection in dynamic scenarios. An extensive experiment confirms that the proposed algorithm outperforms the alternatives in challenging benchmarks. Experimental results also demonstrate that the new algorithm is real-time feasible, given limited computational resources.

3.1 Introduction

Background subtraction is one of the essential tasks in computer vision, which has been utilized in a wide range of applications, including object detection, video surveillance, tracking, video compression and traffic monitoring. Regarded as the first step of most vision systems, background subtraction performance has a heavy impact on the subsequent steps and the overall results. The original idea of background subtraction is simple: a static background frame is available for subtraction from the current frame to obtain any difference in each pixel. If the difference is greater than a given threshold, the pixel is classified as foreground. However, the assumption of the availability of a perfectly static background is almost never found in real-world applications. What we obtain from a pixel is highly dynamic evidence, which corrodes performance. Over the past few decades, diverse methods [44] [15] [97] [103] [101] have been proposed to improve the robustness of background subtraction. Nevertheless, some challenges still remain.

The major challenge is in the dynamic evidence from a single pixel, which may be caused by *dynamic background*, *camera jitter*, *camera automatic adjustment*, *illumination change* and other factors. *Dynamic background* appears most in real-world applications, including background objects such as leaves blowing in the wind, a running escalator or a flowing river. Although moving, these objects should still be treated as part of the background. Wind may also cause *camera jitter*. In this situation, most pixels in the frame are subject to a consistent shaking direction, the equivalent of injecting a particular kind of global noise. This leads to false foreground detection without a robust background maintenance mechanism. Similarly, the *camera automatic adjustment* caused by the intelligent function of modern cameras can

cause this type of correlated global noise. There are two types of *illumination changes*, gradual and rapid. Gradual illumination changes are usually related to an outdoor environment, while rapid change is usually due to an artificial light's "on/off" quality and tends to appear in a repeatable pattern. Furthermore, within observations with the above dynamic factors, some challenging foreground characteristics have to be detected as well, such as *camouflage*, where the foreground looks similar to the background, and *foreground aperture*, where the foreground object has a uniform color region, leading to detection loss in that region.

To address these challenges, a number of approaches to improve background subtraction have been considered. One of the most important is the multi-component strategy exemplified in the Gaussian mixture model (GMM) [103]. This mechanism makes use of multiple components to represent different evidence from a single pixel. Nevertheless, there are some specific drawbacks to most multi-component methods. First, although these methods claim to take advantage of using multiple components to represent multiple evidence, the correspondence between component and evidence is inconclusive, since there is no accurate mapping mechanism to clearly identify relationships. For example, in some conditions, the evidence that is supposed to be updated into the leaf component would also pollute the air component. A non-parametric Bayesian technique is applied to ameliorate this problem, however it is still limited by the hidden density of evidence from the single pixel, which shows a long-tail effect. If something from the foreground occasionally moves into this pixel, the component that is supposed to be created might be submerged by other existing ones, which in turn would cause them to deteriorate. Essentially, traditional per-pixel multi-component methods do not present the real objects in background,

but rather approximate the mixed result of their appearing in the single pixel. The second drawback to multi-component methods is that, as a result of the fuzzy representation, setting the component's variance has been found to be a trade-off. The variance needs to be set high enough to capture the variation in the dynamic nature of the background caused by the challenges described above. But, some foreground components can be detected only when the variance is small: for example, *camouflage* looks like it belongs to the background, either deliberately or accidentally. Moreover, the update process tends to flatten the component.

To this end, a probabilistic topic model method is proposed to estimate a block of pixel densities. Taking advantage of hierarchical Bayesian analysis, this probabilistic model is widely used to uncover the underlying semantic structure of collected materials in the data mining field. For example, it analyzes the words of training documents to infer what the topics are, as well as how the topics are connected to each other. In the background subtraction context, a "real pixel" anchored with a background object is modelled as a topic. A collection of pixel observations, expressed by RGB values, in a certain period is treated as a document. The observed materials from a block of pixels accumulated in the period are the training set. In this way, all the objects in the background can be represented by a complete set of topics and the background's dynamic nature is simulated by the proportion of each topic within each pixel/document. For example, a fluttering leaf that emerges with a tree or obstructs the sky only changes the proportions of different documents accordingly. Not only can this model estimate each topic with a stable mean and variance, but it can also rapidly adapt to emerging new dynamic patterns in the background, as only proportions need to be updated. Furthermore, of great importance for real-world

application, a new topic can be automatically created when enough new instances of evidence become apparent. Since different pixels can share the same set of topics, the proposed algorithm maintains a lower computational cost than most per-pixel approaches.

However, the classic topic model is impeded by two issues that directly apply to background modelling. First, the classic learning methods cannot meet the requirements of on-line style background subtraction. Neither of the two major methods, sampling-based or variational-based, are feasible due to their time consuming nature. Most models are off-line and based on a pre-selected number of topics, a typical design for data mining applications. To handle these problems, a novel inferring algorithm is proposed, which is fully on-line and computationally economical. Second, the number of topics is usually set before the model is learned, which fails in the background subtraction context. In this scenario, if a new topic appeared, it might disappear into the foreground or emerge in the background model.

Thus, to make the topic model suitable to background subtraction, a dynamic topic set mechanism is employed. The challenge is to reveal a new topic as rapidly and accurately as possible based on temporal information that improves the performance for the instance. To this end, each topic is represented by a single Gaussian distribution with no parameter representing prior, as the prior would decrease the influence of the individual piece of evidence and cause unnecessary computational cost. The topic proportion is re-interpreted by "capping-counts" rather than the classic "pseudo-counts", so that if the background topic never appears on a pixel it will not have a larger proportion than a newly emerging foreground topic.

In sum, the contributions of this proposed work are: 1) A probabilistic topic

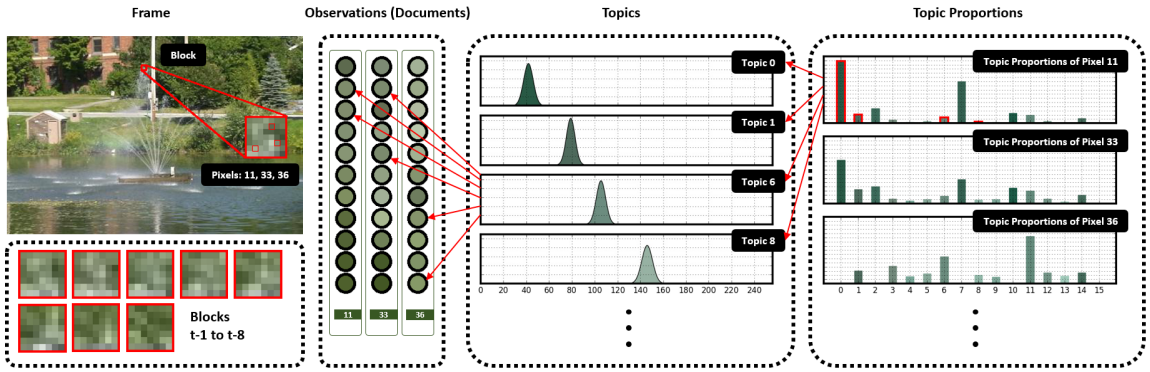

Figure 3.2: The dynamic background is represented by proposed probabilistic topic model

model is first applied to the background subtraction field. A novel background representation based on this model is proposed, robustly depicting the dynamic nature of the background. 2) A novel on-line learning algorithm is proposed, with a lower computational cost than existing methods. 3) The classic topic model is practically reformulated to fit the background subtraction application.

3.2 Related Work

Based on the independently operational cell, background subtraction literature can be summarized into three groups: pixel-based methods, block-based methods, and methods operating the whole frame. The most relevant approaches within the first two groups will be discussed below. For a comprehensive review, readers are referred to the survey papers [89] [34] [14] [15].

3.2.1 Basic Components

The author begin with a review of the literature related to models for a single component. The most famous method is the GMM [103], where each pixel density is estimated by a weighted Gaussian component. Both the foreground and background densities are modelled using this method. This kind of methods is based on the idea that the majority of components should be background and the remainder foreground [43]. Although these Gaussian type components could handle dynamic scenarios, this method presents a number of disadvantages. For example, research shows that the Laplace model is more similar to the pixel density of an indoor scene than a Gaussian component [54]. To mitigate the constraint of a strict Gaussian component, [4] makes use of a general Gaussian component for background modelling. Student-t distribution is also applied to this field by some researchers because of its long-tailed nature [84] [42], which shows robustness to outliers. From the same perspective, a number of non-parametric approaches use kernel density estimation to formulate the background, for example, [33]. There are two well known methods in this sub-field, the Visual Background Extractor [12] and the Pixel-Based Adaptive Segmenter [49]. The first builds components by accumulating single pixel observations, which ensures an exponentially decaying process by a random selection method. The second also makes use of the history of recent observations and employs two per-pixel thresholds to fit the dynamic background. Hybrid methods, which fuse the non-parametric method and the parametric method [30] [72] [73], have also been proposed in recent years. Usually, the combination of a non-parametric regional model, like kernel density estimation, and a parametric model, like GMM, is employed to approximate the background density. In this framework, the algorithm can handle both foreground

detection and shadow removal simultaneously.

3.2.2 Multiple Components

Most existing methods employ more than one component, no matter the type of components discussed before they used. In the original GMM [103], the component count is fixed over time, which is not optimal. An expectation-maximization method is applied in [106] to make GMM adaptive to diverse complexities of different pixels. A training process is applied to estimate the optimal number of components in [17]. To solve the problem, an on-line algorithm for the proposed Dirichlet mixture model is also utilized to estimate the parameters and select the Gaussian count simultaneously in [46]. Similarly, this problem is also handled by probabilistic regularization in [43] and by a non-Gaussian process in [36].

Beyond determining the number of components, three issues related to multiple components arise when a system is designed. The first is the computational cost. Like on-line configuration of the number of components, many improvements would increase this cost, preventing real-time implementation. To mitigate this effect, the GPU technique is used to speed up these algorithms in [43]. To more efficiently update the component parameters, a recursive filter is applied in [84]. A variational Bayes framework based on a batch algorithm is adopted in [36]. In general, computational cost is highly correlated with the number of components maintained on-line for each pixel.

The second issue is related to component weights, which are initially used to weight the sum of the probabilities. In the original GMM, the weights are constantly updated. To improve this model, most researchers focus on adapting or optimizing

the weights, using diverse methods [44] [83]. Some approaches extend the functionality of weights to support classification, so that an additional counter is associated with each weight to determine whether the component belongs to the background [69]. In Dirichlet process based methods, the weights are formulated as multinomial distributions [36]. A classic approach to update this Dirichlet-multinomial parameter is to treat it as a pseudo-count; however, from a practical standpoint, this is unsuitable to background subtraction. In [43], a confidence capping method is used to improve the update process, where an on-line density estimate to selectively forget old components is designed to allow new observations to dominate.

Finally, the third issue related to working with multiple components is the learning rate, which controls the adaptation rate between new evidence and historical data. This rate is a constant in some methods, which slows down the initialization phase and has been shown to be sensitive to environment changes. Another approach employs an adaptive learning rate [126] [140] [49] [59]. For instance, [126] attempts to design a system that automatically adjusts the learning rate. [58] uses a different learning rate for different components. Many cues are investigated to support the rate update, such as spatial information [118], illumination change [91], feedback information [90], and foreground speed/size [34]. The frequency of observation of each component is the cue most commonly utilized to make the learning rate adaptive, as in [69]. However, robustly adjusting the rate on-line still remains a challenge.

3.2.3 Region-based Methods

Exploring inter-pixel relations in a local region shows promising performance on dynamic backgrounds. A straightforward approach is to build a coarse-to-fine cascade

mechanism, with coarse detection on the region level and a fine process at the pixel level. A number of other techniques are also proposed, such as joint distribution [97], patch classification [67], and a graph-based model [20]. In [76], a method separating the foreground by investigating the distinguished patches is proposed. In [137], subspace learning of video patches is used to address challenging scenarios. In [68], spatio-temporal representation is explored to better express the background. The most relevant method is shown in [19], which proposes a sharable mechanism to execute background subtraction. The evidence in each pixel is matched with all components within the pixel's neighborhood; this experiment shows robustness with respect to *dynamic background*. However, its components and weights are estimated independently, without considering either the spatial or the temporal relationship. Essentially, this method attempts to enhance performance by merging a local area of the classic GMM model.

3.3 Algorithm

3.3.1 Representation

In the proposed system, a single frame is divided into a set of equal size $M = l \times l$ blocks 3.2 (a), each of which is modelled independently by the proposed probabilistic topic model. A topic $\boldsymbol{\theta}_k = (\mu_k, \sigma_k)$ is formally defined as a Gaussian distribution over the intensity space $[0, 255]$, as shown in 3.2 (c). Following traditional terminology, a certain amount of intensities observed on a single pixel in a time period T is defined as a document $\boldsymbol{x}^m$ 3.2 (b), where x_t^m is the intensity evidence at time t. Thus, there are $M = l \times l$ documents in one block. The model assumes that one document could

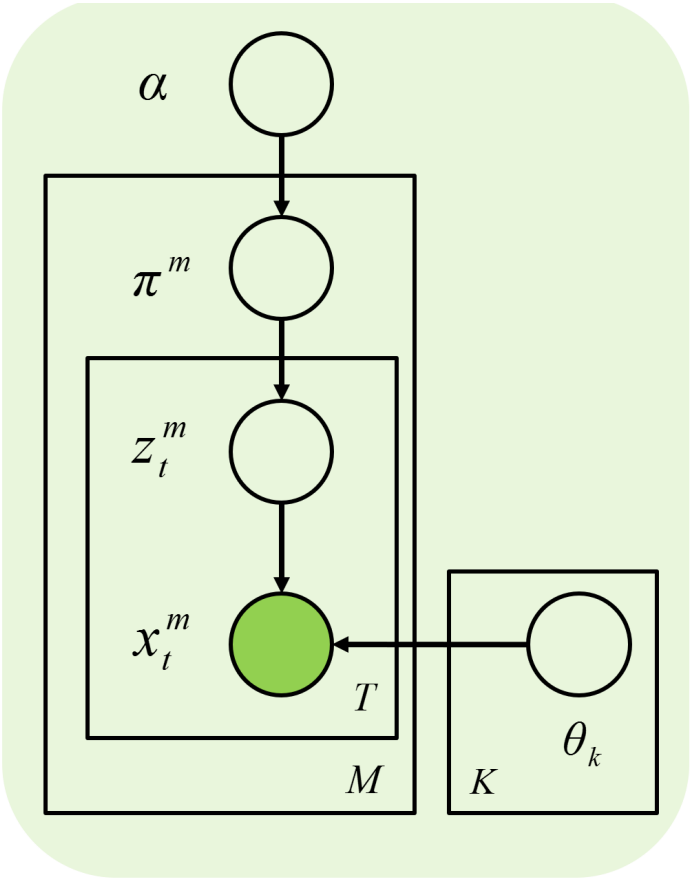

Figure 3.3: The probabilistic graphical representation of the proposed algorithm

exhibit multiple topics. The topic proportion for m-th document (pixel) is $\boldsymbol{\pi}^m$, where π_k^m denotes the topic proportion for the k-th topic, 3.2 (d). In the document m, the topic assignment for the evidence (intensity) x_t^m is expressed as z_t^m; in our context, this will also be called the indicator variable.

This generative process is graphically represented in Figure 3.3. In the graph, the observed variable is every evidence x_t^m in a document; the hidden random variables include the topic proportion $\boldsymbol{\pi}_m$, assignment (indicator variable) z_t^m, topic $\boldsymbol{\theta}_k$, and prior of the proportions $\boldsymbol{\alpha}$. M denotes the number of documents within the collection (block) and T denotes the collection size of the evidence in each document. This graph

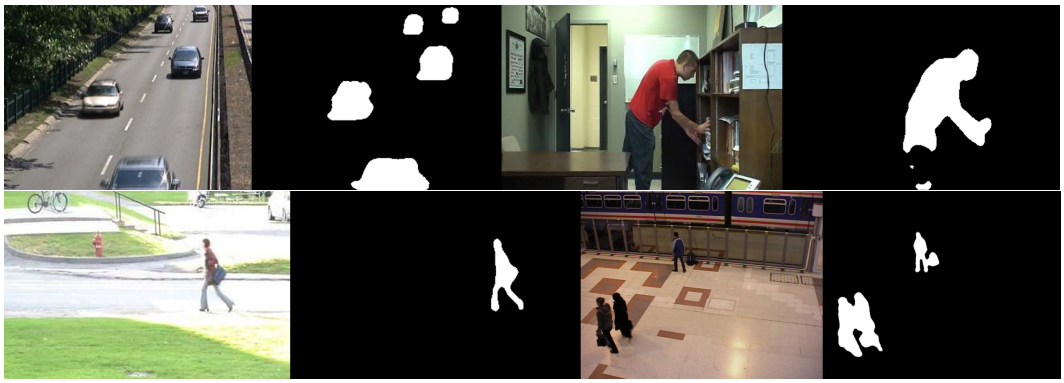

Figure 3.4: Sample results of *baseline* of dataset *CD2014*

expresses the dependency relationships between variables, which can also be expressed by the following distributions:

$$\boldsymbol{\pi}^m|\boldsymbol{\alpha} \sim \mathcal{D}ir(\boldsymbol{\alpha}) \tag{3.2}$$

$$z_t^m|\boldsymbol{\pi}^m \sim \mathcal{M}ul(\boldsymbol{\pi}^m) \tag{3.3}$$

$$x_t^m|z_t^m, \{\boldsymbol{\theta}_k\}_{k=1}^K \sim \mathcal{N}(\boldsymbol{\theta}_k) \tag{3.4}$$

where a evidence x_t^m is drawn by the Gaussian distribution $\mathcal{N}(\boldsymbol{\theta}_k)$ indicated by z_t^m, a topic assignment variable z_t^m is randomly chosen by the corresponding multinomial distribution $\mathcal{M}ul(\boldsymbol{\pi}^m)$ of the m-th document that simulates the topic proportion, and a topic proportion $\boldsymbol{\pi}^m$ is generated by a Dirichlet distribution $\mathcal{D}ir(\boldsymbol{\alpha})$. The observed intensities can be treated as arising from this generative process.

From a background subtraction perspective, this statistical model reflects the underlying insight that the intensity sequence observed from a single pixel (document) could exhibit multiple Gaussian models (topics). All intensity evidence in a block share the same set of Gaussian models. Each pixel position exhibits the Gaussian set in a different proportion. Each intensity from a particular pixel is drawn from

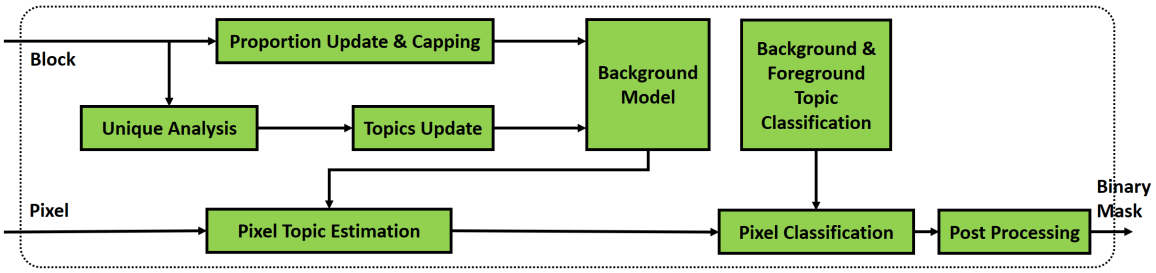

Figure 3.5: On-line learning framework

Figure 3.6: Sample results of *challenging weather* of dataset *CD2014*

one Gaussian distribution such that the selected Gaussian component is chosen from the per-pixel distribution over the Gaussian set (that is, the distribution is anchored to the pixel). Based on the model, this work attempts to collect a stable complete set of Gaussian models, where each Gaussian component represents a semantic part of one and only one real object. For example, the sky pixel has intensities with a high probability for sky and very low probabilities for other objects. Whether or not other objects have appeared on the pixel before, the complete set of the model is stably stored in the system and the pixel could adapt to intensities about other objects quickly, as only its proportions will be updated.

3.3.2 Topic Learning

The goal of topic modelling is to automatically discover hidden structures from a collection of documents. Conventionally, the hidden variables are inferred via sampling-based [92] or variational [108] methods, most of which employ an off-line approach and cannot meet real-time requirements. In the background subtraction context, an incremental learning approach is necessary, since the documents change with every frame and historical data cannot be stored. For each pixel, a stream of evidence arrives, one piece of evidence within each frame. A fixed length document can be seen as a window through the evidence stream. A low computational cost is also critical. In accordance with the characteristics of background subtraction, a novel on-line hidden structure learning method is proposed in this chapter, as shown in Algorithm 2. This algorithm takes a block of new frame evidence $\{x_t^m\}_{m=1}^M$ as input and makes use of them to update topics and topic proportions separately.

To learn the topic parameters, a uniqueness analysis is first applied to the block of intensities $\{x_t^m\}_{m=1}^M$. In this process, redundancy is removed, which means that each intensity value can only appear once in the result. Thus, a new evidence set $\{x_t^j\}_{\boldsymbol{J}^u}$ is obtained, where each element x_t^j is unique $j \in \boldsymbol{J}^u$. The underlying insight is that, since each instance of evidence in the block shares the same set of topics, the correlations among topics and between topic and proportion can be relaxed. One topic is motivated by a single objective, representing the Gaussian distribution over intensity space for that topic only; this leaves only the task of learning weighting information for every document's proportion. From this perspective, the count of evidence with the same intensity provides no information for topic learning.

Before updating the topics, the indicators $\{z_t^j\}_{\boldsymbol{J}^u}$ are estimated. This can be done

Figure 3.7: Sample results of *camera jitter* of dataset *CD2014*

by comparing the densities of x_t^j in each Gaussian distribution, shown in (3.5).

$$z_t^j = \arg\max_{\{k \in \boldsymbol{K}_{t-1}\}} \left\{ \mathcal{N}\left(x_t^j | \mu_{k,t-1}, \sigma_{k,t-1}\right) \right\} \tag{3.5}$$

where $\boldsymbol{K}_{t-1} = [1, K]$ denotes the topic set at time $t-1$; this may vary in the on-line process and will be discussed in the following subsection. Note that Equation (3.5) is functional in the topic learning module because of the removal of count information.

$$\mu_{k,t} = (1 - \sum_{z_t^j = k} w_j \epsilon_j) \mu_{k,t-1} + \sum_{z_t^j = k} w_j \epsilon_j x_t^j \tag{3.6}$$

All the instances of evidence whose indicator variables are equal to k are utilized to update the corresponding Gaussian parameters of the k-th topic. The mean is updated in (3.6). where w_j and ϵ_j are the normalized weight and the learning rate of each evidence j, respectively. The parameter updates a certain ratio for each frame, $\sum_{z_t^j = k} w_j \epsilon_j$, a weighted sum of the learning rates of all collected evidence. If there is only one instance of evidence, (3.6) degenerates to the conventional GMM update

procedure [103]. The normalized weight w_j is calculated as

$$w_j = \frac{\mathcal{N}\left(x_t^j | \mu_{k,t-1}, \sigma_{k,t-1}\right)}{\sum_{z_t^{j'}=k} \mathcal{N}\left(x_t^{j'} | \mu_{k,t-1}, \sigma_{k,t-1}\right)} \tag{3.7}$$

The learning rate ϵ_j is computed by

$$\epsilon_j = \tau \mathcal{N}\left(x_t^j | \mu_{k,t-1}, \sigma_{k,t-1}\right) \tag{3.8}$$

where τ is a constant parameter controlling the rate. Similarly, Gaussian variance is updated in (3.9).

$$\sigma_{k,t}^2 = (1 - \sum_{z_t^j=k} w_j \epsilon_j)\sigma_{k,t-1}^2 + \sum_{z_t^j=k} w_j \epsilon_j (x_t^j - \mu_{k,t})^2 \tag{3.9}$$

where $\sigma_{k,t-1}^2$ denotes the variance of the k-th topic.

Based on the update process, the mean of a topic will oscillate around a little area in the intensity space. The variance tends to shrink starting from the large initial value σ_0 that is assigned as a topic is created. Reducing to a certain value, the variance then appears stable. In addition, topics evolve independent of each other, which alleviates the long-tail effect from a global perspective. Topic creation and deletion will be presented in Section 3.3.4.

3.3.3 Proportion Learning

Assuming a given sequence of estimated indicators $\boldsymbol{z}_{t-1}^m = (z_1^m, ...z_{t-1}^m)$ for document m, what our first concern is the topic proportion's conditional posterior $p(\boldsymbol{\pi}^m | \boldsymbol{z}_{t-1}^m, \boldsymbol{\alpha})$.

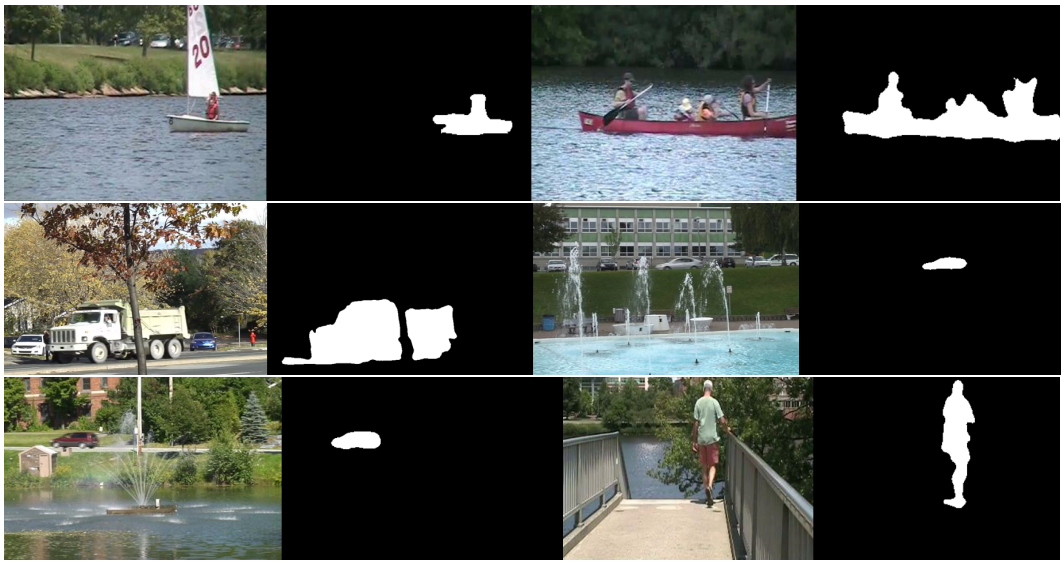

Figure 3.8: Sample results of *dynamic background* of dataset *CD2014*

Because of the conjugate Dirichlet prior's property, (3.10) can be derived.

$$p(\boldsymbol{\pi}^m|\boldsymbol{z}^m_{t-1},\boldsymbol{\alpha}) = \mathcal{D}ir(n^m_1+\alpha_1,...,n^m_K+\alpha_K) \tag{3.10}$$

where $n^m_k = \sum_{t'=1}^{t-1}\delta(z^m_{t'}-k)$ denotes the number of times the evidence in the sequence are assigned to topic k. Since the posterior is still a Dirichlet distribution, the parameter $\boldsymbol{\alpha}^m_{t-1} = (\alpha^m_{1,t-1},....\alpha^m_{K,t-1})$ can be considered to be "pseudo-counts" of the indicator variables.

For the indicator z^m_t of a new incoming piece of evidence, the probabilistic density of z^m_t can be calculated by (3.11).

$$\begin{aligned} & p\left(z^m_t{=}k|\boldsymbol{z}^m_{t-1},\boldsymbol{\alpha}\right) \\ &= \int p(z^m_t{=}k|\boldsymbol{\pi}^m)p(\boldsymbol{\pi}^m|\boldsymbol{z}^m_{t-1},\boldsymbol{\alpha})d\boldsymbol{\pi}^m \\ &= \frac{\alpha^m_{k,t-1}}{\sum_{k'=1}^{K}\alpha^m_{k',t-1}} \end{aligned} \tag{3.11}$$

Figure 3.9: Sample results of *intermittent object motion* of dataset *CD2014*

where the topic proportion $\boldsymbol{\pi}^m$ is integrated out. As a result, the only parameter needing to be on-line updated is "pseudo-counts" $\boldsymbol{\alpha}^m$.

For the newly provided evidence x_t^m, the value of the associated indicator z_t^m is determined before updating the $\boldsymbol{\alpha}_{t-1}^m$. The details of determining z_t^m will be presented in Section 3.3.5. At this point, one has

$$\alpha_{z_t^m,t}^m = \alpha_{z_t^m,t-1}^m + \epsilon_{z_t^m} \tag{3.12}$$

where only the z_t^m-th element of $\boldsymbol{\alpha}_{t-1}^m$ increases, while the other elements remain the same. To take the confidence into account for updating, the learning rate of this evidence $\epsilon_{z_t^m}$ (3.8) is utilized rather than the counter 1.

To adapt to long-term application, a capping mechanism is used to tune the updating by checking each element after the update. If an element k is greater than a cap C, a multiplier $\frac{C}{\alpha_{k,t}^m}$ will be applied to every element of $\boldsymbol{\alpha}_t^m$. This mechanism works

Algorithm 2 On-line Learning Algorithm

Input: $\{x_t^m\}_{m=1}^M$, $\{\boldsymbol{\theta}_{k,t-1}\}_{k=1}^{K_{t-1}}$, $\{\boldsymbol{\alpha}_{t-1}^m\}_{m=1}^M$;
Output: $\{\boldsymbol{\theta}_{k,t}\}_{k=1}^{K_t}$, $\{\boldsymbol{\alpha}_t^m\}_{m=1}^M$;

uniqueness analysis and obtain $\{x_t^j\}_{\boldsymbol{J}^u}$
estimate $\{z_t^j\}_{\boldsymbol{J}^u}$ (3.5)
calculate $\{\epsilon_j\}_{\boldsymbol{J}^u}$
calculate $\{w_j\}_{\boldsymbol{J}^u}$
for $k \in \boldsymbol{K}_{t-1}$ **do**
    update $\mu_{k,t}$
    update $\sigma_{k,t}$
end for
for $m \in [1, M]$ **do**
    estimate z_t^m
    update $\boldsymbol{\alpha}_t^m$
end for
create and delete topics

by limiting how high of the "pseudo-counts" can go, which prevents any element from unlimited increase.

3.3.4 Dynamic Topics

The literature has shown that the fixed topic count cannot adapt to the complications of real-world background subtraction. Thus, a dynamic topic count is applied in the learning process. Note that pursuing theoretical completeness is not the goal of this work, as that usually needs to take the Gaussian parameter prior into account, which tends to be computationally inefficient and create more parameters for tuning. In this algorithm, a straightforward method is used.

In the on-line learning process, $\boldsymbol{K}_{t-1}=[1, K]$ is defined as the topic set at time $t-1$, which includes both background and foreground topics $\boldsymbol{K}_{t-1} = \boldsymbol{K}_{t-1}^+ \cup \boldsymbol{K}_{t-1}^-$. If a new unique instance of evidence x_t^j generated from the uniqueness analysis module lies

Figure 3.10: Sample results of *low frame-rate* of dataset *CD2014*

outside of all two-standard-deviation areas of the whole topic set in intensity space, a new topic will be created with a mean of x_t^j and an initial variance σ_0^2. This creation criteria means Equation. (3.13) holds for all k.

$$|x_t^j - \mu_{k,t-1}| > \psi\sigma_{k,t-1} \tag{3.13}$$

where $|\cdot|$ is an absolute operator and $\psi = 2$ is a constant. A newly created topic is marked as foreground. A similar mechanism is used in the GMM method.

For each existing topic, a counter is utilized to measure the topic's status. If there is a new unique instance of evidence x_t^j that is estimated to belong to a foreground topic k, $z_t^j = k$, this topic is marked as visited and the corresponding counteris incremented by 1. A different procedure is used for a background topic: the counter is immediately assigned the maximum counting value $\mathcal{C}_c$. If there is no evidence belonging to the topic, for both background and foreground, it is considered to be unvisited and the corresponding counter is decremented by 1. If the value is smaller than 0, a deletion process will be triggered. Based on this method, background topics will have a tolerance interval to be deleted and foreground topics will oscillate between $(0, \mathcal{C}_c]$.

Figure 3.11: Sample results of *night* of dataset *CD2014*

Like most background subtraction methods, the algorithm's determination of whether a topic belongs to the background relies on the topic's existing time. This measurement is logical because the essential meaning of background is that an object stays in one place for a long enough time. For each topic, a time-accumulator t'_k is used, which will increase 1 per frame as long as this topic k exists, and is not deleted. If the accumulator is greater than a period threshold $\mathcal{P}_b$, the topic is marked as background.

In this way, another advantage of the proposed topic model is revealed. For each pixel position, the topic proportion of the topics and the sign of background/foreground is purely separated, meaning that those background topics that could potentially appear but rarely do appear can be stably included in the model.

3.3.5 Pixel Classification

Given an instance of evidence x_t^m on the m-th pixel, the likelihood $p(z_t^m{=}k, x_t^m|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}^m, \{\boldsymbol{\theta}_1^K\}, \boldsymbol{\alpha})$, that the evidence is generated by the k-th topic can be calculated based on the fore-learned model (3.14).

$$\begin{aligned} & p\left(z_t^m{=}k, x_t^m|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}^m, \{\boldsymbol{\theta}_k\}_1^K, \boldsymbol{\alpha}\right) \\ &= p(z_t^m{=}k|\boldsymbol{z}_{t-1}^m, \boldsymbol{\alpha}) p(x_t^m|z_t^m{=}k, \boldsymbol{x}_{t-1}, \{\boldsymbol{\theta}_k\}_1^K) \\ &= \frac{\alpha_{k,t-1}^m}{\sum_{k'=1}^K \alpha_{k',t-1}^m} \mathcal{N}(x_t^m|\mu_{k,t-1}, \sigma_{k,t-1}) \end{aligned} \tag{3.14}$$

where the second term is derived by the Markov property and Bayes' rule. The Markov property states that the conditional probability distribution of future states of the stochastic process depends only upon the present state, not on the sequence of events that preceded it. For example, in this Bayesian network of Figure. 3.3, the new observed variable x_t^m is independent of its non-descendant $\boldsymbol{\alpha}$ given its parents, which simplifies the latter probability of the second term. The final likelihood result equals the k-th topic proportion multiplied by the probability of the k-th topic, given the evidence x_t^m. The value of z_t^m is chosen by maximizing the above likelihood (3.15).

$$z_t^m = \underset{k' \in \boldsymbol{K}_t}{\operatorname{argmax}} \{p(z_t^m{=}k', x_t^m|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}^m, \{\boldsymbol{\theta}_k\}_1^K, \boldsymbol{\alpha})\} \tag{3.15}$$

Note that the above Equation (3.15) is utilized in both the pixel topic estimation and the proportion update & capping modules, shown in Figure 3.5.

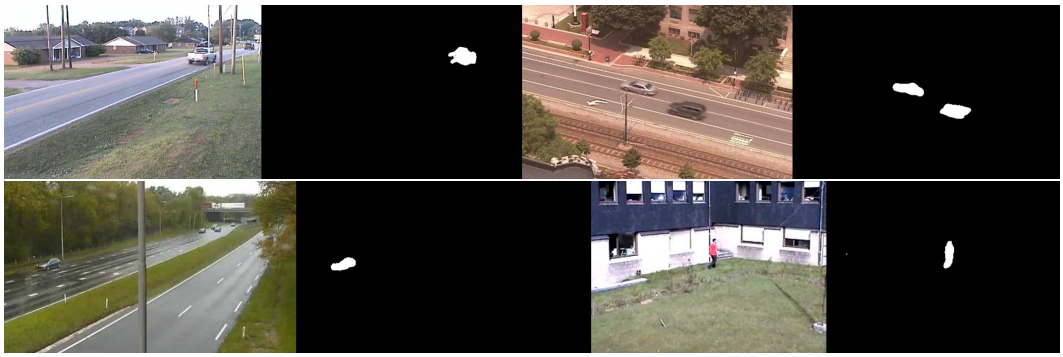

Figure 3.12: Sample results of *PTZ* of dataset *CD2014*

Finally, the indicator variable z_t^m can be determined. One has

$$\begin{aligned} & p(x_t^m|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}^m, \{\boldsymbol{\theta}_k\}_1^K, \boldsymbol{\alpha}) \\ & = \sum_{z_t^m=1}^{K} p(z_t^m, x_t^m|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}^m, \{\boldsymbol{\theta}_k\}_1^K, \boldsymbol{\alpha}) \end{aligned} \tag{3.16}$$

where $p(x_t^m|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}^m, \{\boldsymbol{\theta}_k\}_1^K, \boldsymbol{\alpha})$ is the marginal likelihood of x_t^m, given the learned model.

The algorithm's essential goal is to determine whether the evidence x_t^m belongs to the foreground. The evidence will be classified as foreground if one of the two cases shown below is satisfied. Otherwise, it is background. The first case is that this new evidence is classified as a foreground topic.

$$z_t^m \in \boldsymbol{K}_t^- \tag{3.17}$$

The second case is that the marginal likelihood of the evidence is lower than a given threshold.

$$p(x_t^m|\boldsymbol{x}_{t-1}, \boldsymbol{z}_{t-1}^m, \{\boldsymbol{\theta}_k\}_1^K, \boldsymbol{\alpha}) < \mathcal{C}_{th} \tag{3.18}$$

Figure 3.13: Sample results of *shadow* of dataset *CD2014*

which means that a new foreground topic is emerging. Note that the parameter $\mathcal{C}_{th}$ does not need to be tuned, as it is always kept as a near-zero scalar for all the scenarios.

3.4 Implementation Details

The core of the proposed algorithm has been presented above, but some implementation details remain.

Initialization. The proposed model is initialized with the first input frame, which may contain foreground objects. For each block, the topics are constructed by making use of the unique pieces of evidence to feed a Gaussian mixture model, based on the expectation-maximization algorithm.The counter of each topic is assigned as 1. All topics are marked as foreground. At the beginning, all topics represent the foreground, but may include background objects. As background topics are observed

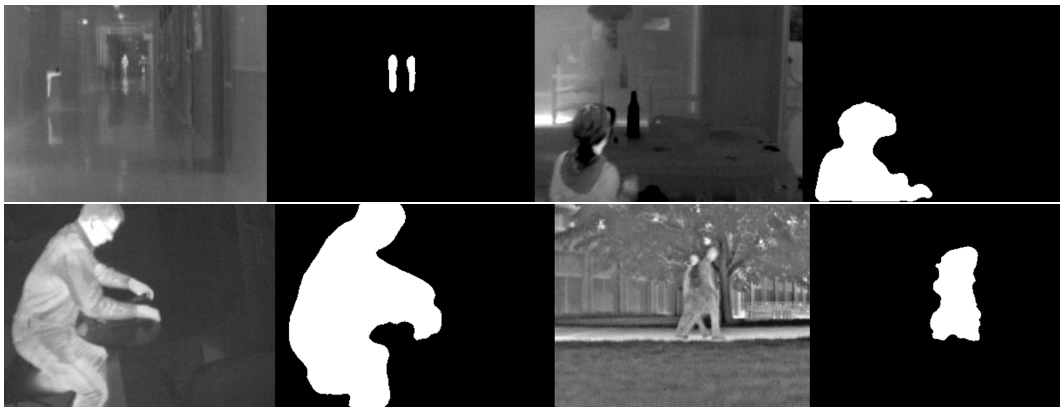

Figure 3.14: Sample results of *thermal* of dataset *CD2014*

stably in sequence, the time-accumulator of a background topic will eventually exceed the threshold $\mathcal{P}_b$. As foreground objects move away, the counter of a foreground topic will eventually drop down to 0 and be deleted. Ultimately, the topic set would tend to be stable.

Channel Fusion. All of the descriptions of the proposed algorithm so far have been based on a single channel frame. To handle color frames, the three channels need to be fused. In this system, if a pixel is detected as foreground in one of the three channels, the position is considered to belong to the foreground. The underlying logic is that the stable and complete topic set should be able to consistently represent the background evidence in every channel. From the foreground perspective, if one of the RGB values of a pixel exceeds the domain of the corresponding topic set, it should be considered to be foreground.

Post Process. An alternative phase exists between the pixel classification and the final binary mask: the post process module. This additional module is provided to improve the output of the background model from different perspectives. For example, the foreground decision may be based only on a single pixel or local area information,

rather than the whole frame. A number of works in the background subtraction literature include this improvement phase, but it is methodologically independent of the background modelling. In this chapter, no sophisticated method is applied in this phase: only a small blob removal operation is utilized. Specifically, a connected component analysis is first applied to identify all the blobs. Then, the small blobs constructed by little pixels are ignored, since they are usually caused by noise.

Parameters. This background model includes a Dirichlet prior $\boldsymbol{\alpha}_0^m$ on the multinomial parameters for each document, which generally has a smoothing effect on the multinomial distribution. In this chapter, a symmetric prior is used, that is, the assumption of the model with respect to priors is that all topics have the same chance of being assigned to a document. For topic parameters, a initial variance σ_0 is used as described above, since no information on priors is applied. A minimum value for the variance $\sigma_{k,t}$ is also bounded with value 0.5, which ensures that a topic can have at least dominant 3 intensities in the space. To control the learning rate of each update, the parameter τ is set as 0.005. The size of blocks is fixed with pixel length of 7. The capping value for topic counters is set as 50. The threshold for the likelihood of each new piece of evidence is set a very small scalar, $1.0e^{-7}$. To implement the dynamic topic system, a topic pool with maximum size $K^{max} = 100$ is utilized. All

σ_0^2	initial variance of a new topic	25.0
τ	parameter controlling learning rate	0.005
l	length of block	7
$\alpha_{k,0}^m$	Dirichlet prior	1.0
$\mathcal{C}_c$	maximum value of topic counter	50
$\mathcal{C}_{th}$	probability threshold	$1.0e^{-7}$
K^{max}	size of the topic pool	100

Table 3.2: Parameters and corresponding values of the background subtraction system

the described parameters are summarized in Table 3.2, and are fixed in every tested benchmark.

3.5 Experiments

The experiments are conducted on three public benchmarks, namely *CD2012*, *CD2014*, and *SABS*. This extensive experiment enables the proposed algorithm to be compared with a number of state-of-the-art methods. Unlike traditional datasets, which are either too small or not challenging enough for recently published approaches, these three datasets contain a wide variety of difficult sequences. Furthermore, these sequences are separated into different challenging categories, which makes it easier to analyze the algorithm's performance. The sequences from *CD2012* and *CD2014* are obtained from real-world situations. To remove the effect of the environment, *SABS* offers a synthetic evaluation benchmark, consisting of a 3D virtual world. Together, there are more than $160,000$ annotated frames tested in this experiment.

To quantitatively evaluate the algorithms, three metrics are utilized, namely *Recall* Re, *Precision* Pr, and *F-measure* Fm. *Recall* measures the rate of true positives out of the total in the ground truth.

$$Re = \frac{Tp}{Tp + Fn} \tag{3.19}$$

where Tp and Fn are the total number of true positives and false negatives respectively. *Precision* is defined as

$$Pr = \frac{Tp}{Tp + Fp} \tag{3.20}$$

Method \ Task	Baseline	Jitter	Dynamic	Intermittent	Shadow	Thermal
Shared GMM [19]	0.935	0.817	0.867	0.798	0.813	0.825
FTSG [121]	0.933	0.751	0.879	0.789	0.883	0.777
MBS V0 [95]	0.928	0.836	0.790	0.709	0.778	0.811
AMBER+ [115]	0.881	0.711	0.843	0.721	0.813	0.760
EFIC [3]	0.917	0.713	0.578	0.578	0.820	0.838
AAPSA [93]	0.918	0.721	0.671	0.510	0.795	0.703
SuBSENSE [100]	0.950	0.815	0.818	0.657	0.899	0.817
MD [13]	0.895	0.496	0.526	0.496	0.568	0.706
Proposed Algorithm	0.940	0.710	0.880	0.766	0.891	0.827

Table 3.3: Performances of compared algorithms on *Change Detection 2012* (Dynamic: *dynamic background*, Jitter: *camera jitter*, Intermittent: *intermittent object motion*)

where Fp is the number of false positives. The *F-measure* is calculated as the harmonic mean of *Recall* and *Precision*,

$$Fm = \frac{2 * Re * Re}{Re + Pr} \tag{3.21}$$

The F-measure is the major measurement utilized to rank the methods under comparison.

The algorithm is implemented in C/C++. The experiment runs on a Core i7 CPU running at 3.5 GHz with 12 GB of memory. Currently, this implementation is not optimized. For a color frame with resolution 320×240, the system can obtain 21.2 frames per second based on a single core.

3.5.1 Change Detection 2012

This dataset can be divided into six challenges, which vary from *baseline*, *dynamic background*, to *camera jitter*, *intermittent object motion* and *thermal*. Pixel-specific annotations about background and foreground as well as shadows and their boundaries can be found in the ground truth. For some video clips, a restricted image

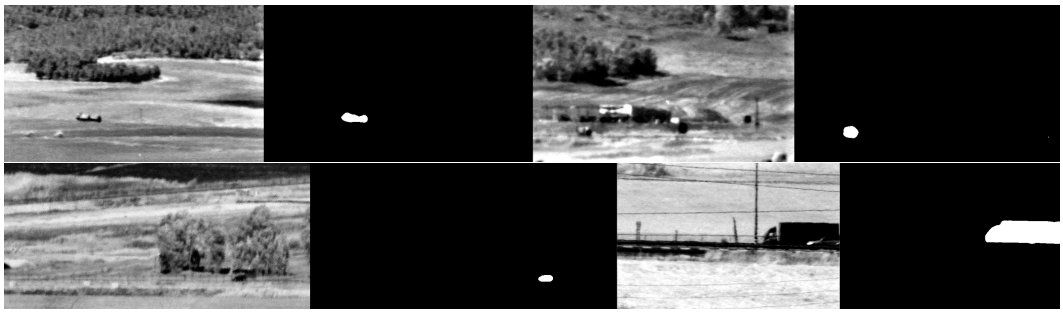

Figure 3.15: Sample results of *turbulence* of dataset *CD2014*

region is utilized only for evaluation. A certain number of frames is provided for each sequence for the training process, if needed. With the dataset, a complicated ranking mechanism is utilized to compare alternative algorithms. In this chapter, the F-measure is used instead, as it is the most commonly utilized metric.

Table 3.3 shows an F-measure comparison for per-category performance with other 8 methods, which are specified in the table. The sampled results of *baseline* can be found in Figure 3.4. This result shows that the proposed method outperforms other methods, in general. The proposed method ranks within the top two in four out of six cases, which provides a better balance between *Precision* and *Recall*. The results demonstrate that this proposed probabilistic topic model successfully represents the characteristics of real-world backgrounds, which can be especially supported by the tested sequences from the *dynamic background* column. In particular, the assumption that the pixels in a local block indeed share a common set of topics and the dynamic nature of background can be reflected by the proportions of the topic set is shown to be correct. Furthermore, the proposed algorithm contributes to the progress of the state-of-the-art performance of background subtraction.

Table 3.3 also indicates that the results of this work are slightly worse than the first-ranked methods in both the *intermittent motion* and *camera jitter* categories.

Method \ Task	Challenging Weather	Low Frame-Rate	Night	PTZ	Air Turbulence
SuBSENSE [100]	0.859	0.651	0.559	0.348	0.779
FTSG [121]	0.822	0.626	0.513	0.324	0.712
MBS V0 [95]	0.773	0.569	0.516	0.512	0.570
EFIC [3]	0.778	0.663	0.655	0.584	0.671
AMBER+ [115]	0.767	0.469	0.380	0.135	0.755
Proposed Algorithm	0.815	0.658	0.415	0.592	0.785

Table 3.4: Performances on *Change Detection 2014*

Our thought is that these two challenges need particular process modules from a background modeling perspective. The first, *intermittent object motion*, contains two aspects, injecting and removing objects, both of which require a faster adaptive speed for labeling a topic as background. However, the speed cannot be too high, because real foreground topics would then be incorrectly labeled. Clearly, there is a trade-off, one is hard to improve without modeling the foreground. This category is usually considered to be the hardest challenge in the literature. For the other category, *camera jitter*, our performance will be improved if a jitter detection module is added. In the table, most of the compared approaches have employed extra sophisticated methods to handle shadows, resulting in their higher scores.

Another advantage of the proposed model can be seen when it is compared with the shared GMM method [19] in the *dynamic background* category, which similarly makes use of a set of Gaussian components, estimated by the evidence collected from not only a single pixel but also a local block. Table 3.3 demonstrates that the shared GMM performs better than most alternative methods. This result validates, from a different perspective, that modeling the surrounding intensities—potentially moving into the target pixel—will improve performance. However, the shared GMM method performs worse than the proposed method, because the shared GMM tends to only make use of the topic set, ignoring the topic proportions.

3.5.2 Change Detection 2014

The 2014 dataset is more complex than the original version. More challenges are included, such as bad weather conditions, low frame-rate sequence, a bad illumination situation, and pantilt-zoom operations. The details of the challenges can be found in Table 3.4. Other configurations are kept the same as those in the *CD2012*. For this dataset, there are 5 methods compared, using the F-measure metric. The sampled results can be found in Figure 3.4 and Figure 3.6 to Figure 3.15.

Table 3.4 shows the quantitative performance comparison for the new challenge categories. From the results, the more challenging character of this new version dataset is clearly apparent. For all algorithms, the *challenging weather* category is the only category showing results comparable to the former dataset. The F-measures drop significantly in all other new challenges. The *PTZ* category violates the basic assumption of background subtraction, that there is no dramatic movement of the background objects. The *night* and *low frame-rate* video sequences can be characterized as the most challenging ones. The proposed method tends to be sensitive to the kinds of noise arising from these challenges, because this noise shows repeatability in the sequences. According to our analysis, these two challenges can be handled by a fore-process module. In general, the proposed algorithm remains the first-ranked approach for the entire dataset as well.

Interestingly, the proposed method shows robustness in the *air turbulence* category, while other algorithms' performances diminish. This means the probabilistic topic model is sufficient for this condition. The reason for this finding is that there is a clear pattern in the *air turbulence* category, which appears similarly for the whole

Method\Task	Basic	Dynamic	Bootstrap	Darkening	Light	Night	Camouflage	No Camouflage	H.264
McFarlane [79]	0.614	0.482	0.541	0.496	0.211	0.203	0.738	0.785	0.639
Stuffer [103]	0.800	0.704	0.642	0.404	0.217	0.194	0.802	0.826	0.761
McKenna [80]	0.522	0.415	0.301	0.484	0.306	0.098	0.624	0.656	0.492
Li [65]	0.766	0.641	0.678	0.704	0.316	0.047	0.768	0.803	0.773
Zivkovic [140]	0.768	0.704	0.632	0.620	0.300	0.321	0.820	0.829	0.748
Maddalena [75]	0.766	0.715	0.495	0.663	0.213	0.263	0.793	0.811	0.772
Barnich [12]	0.761	0.711	0.685	0.678	0.268	0.271	0.741	0.741	0.799
Proposed	0.842	0.831	0.644	0.676	0.500	0.310	0.838	0.858	0.805

Table 3.5: Performances on *Stuttgart Artificial Background Subtraction*

scene. Particularly, some new intensities generated by the combination of the turbulence and real object's color is correctly captured by the algorithm's creating new topic mechanism. Furthermore, turbulence's movement is appropriately depicted by the topic proportions.

3.5.3 Stuttgart Artificial Background Subtraction (SABS)

The SABS is a synthetic dataset created specifically for evaluating background subtraction algorithms. It consists of a 3D rendering of a road junction, shown in Figure 3.16. The foreground objects include cars and people. The whole dataset can be categorized into 9 challenges, namely *basic*, *dynamic background*, *bootstrap*, *darkening*, *light switch*, *noisy night*, *camouflage*, *no camouflage*, and *H.264*. The advantage of making use of a synthetic dataset is its purity, as it can remove the effect of uncontrolled environmental noise. For example, in the *dynamic background*, researchers can focus on testing the algorithm's performance for a swaying tree, since all other surroundings remain the same as the *basic* scene. Similarly, in the *light switch*, reseachers' attention can be focused exclusively on the store switching its light on.

Table 3.5 shows the quantitative performances. The sampled results of our method

Figure 3.16: Sample results of dataset *Stuttgart Artificial Background Subtraction*

can be found in Figure 3.16. In general, the proposed method shows results comparable to those of the state-of-the-art algorithms. The three worst categories are *darkening*, *night* and *bootstrap*. For the *darkening* scene, the sensitive performance is caused by the stability of our topic set. In most multiple component based approaches, the mechanism makes the existing components adapt to the different parameters. This process is controlled by a pre-set learning rate. For the proposed algorithm, the components are more stable. The mechanism changes to creating new topics to cover the situation, which requires more computational time than other methods, and consequently lowers the score. But eventually the background will convert to the new environment. On the other hand, this method also gains a more stable background in common situations. Similarly, in *bootstrap*, the algorithm needs more time to build a stable set of topics. In *night*, different kinds of noises are superimposed into the video, affecting the learning process of the proposed model. In practice, this noise can be removed by a fore-processor.

The advantages of the proposed algorithm are demonstrated in the *dynamic background*, *camouflage*, and *H.264* categories. The very nature of the proposed topic model for handling a dynamic background has been demonstrated, which is also consistent with the experimental results in Table. 3.3 and Table. 3.4. For the *camouflage*, making use of an indicator variable for each item of evidence ensures that one topic

update process will not affect others, which makes the variance smaller and stabler than traditional methods. As a result, the proposed algorithm detects the foreground better in the *camouflage* situation. Although the content quality of *H.264* is degraded, the results show that this method still works appropriately. This in turn shows that a foregoing noise removal process will improve performance in the *night* and *low frame-rate* of *CD2014*.

3.6 Conclusion

In this chapter, the author proposed a novel background subtraction method based on a probabilistic topic model. Taking advantage of a newly designed on-line learning algorithm, the topics and topic proportions of the model were efficiently updated incrementally. This model kept a set of stable Gaussian topics representing the real background objects rather than the instant evidence and flexible topic proportions to reflect the movement of background objects. As a result, the method suitably modeled the motion of dynamic backgrounds and outperformed state-of-the-art methods on multiply challenging datasets.

The proposed method relies on the observation that the evidence from a block of neighbor pixels is sharably generated by a set of real topics. The topic model infers the hidden topics and the topic proportions anchored to each pixel in an online manner. The proposed method can be seen as an extension of the classic GMM and theoretically reveals the connection between the independent per-pixel model and the block model. On a practical level, the proposed method solves the stubborn problem of classic multi-component methods, the long-tail nature of per-pixel density.

Chapter 4

Airborne Ship Detection by Maritime Background Modelling

Airborne maritime surveillance has attracted a great deal of attention in surveillance literature. Extracting objects of interest from the ocean is usually considered to be the first step of the surveillance platform. In this chapter, a novel airborne ocean background modeling method is presented to solve this task. The input of this system is a single image captured by an airborne sensor, while the output is a binary mask indicating extracted objects. To represent the ocean background, each pixel of the image is modeled by a Gaussian Mixture Model, whose component parameters are shared across the scene and whose component weights are dominated by a set of spatial distributions over the image plane. This spatial distribution is designed to tolerate changes in the ocean's texture. An online learning method is employed, which not only estimates the parameters of the distributions but also infers the number of components. An intensive experiment is conducted on a large number of images to demonstrate the proposed algorithm's robustness and accuracy for modeling ocean

background.

4.1 Introduction

Ocean background modeling from the airborne view is an important technique for maritime surveillance platforms. As the first phase of this type of surveillance system, it is a critical element for further analysis, like early understanding and prediction of abnormal shipping behavior. In this chapter, the input is defined as one single image captured by a sensor mounted on an aircraft. The output is the binary mask of this image, denoting whether each pixel belongs to the object foreground or the ocean background.

Although the proposed method shares the same motivation with traditional background subtraction approaches [101], that is, extracting objects of interest by modeling the background, the traditional approaches are not applicable to this chapter's task. For one, the most important assumption of background subtraction—that the scene is static and the object is moving—fails. In airborne maritime surveillance, the aircraft is always moving while the object of interest may not significantly move in the image. For another, high-frame-rate video may not always be available in the maritime surveillance application. Different with the background subtraction, making use of the majority in time domain, this proposed work explores the majority in image domain based on the assumption that ocean texture appears similar in a single image.

Another related technique is salient object segmentation [130] [37] [132] [1]. The proposed work can be considered a salient object extraction method that models the ocean background. However, airborne maritime surveillance has its own distinct

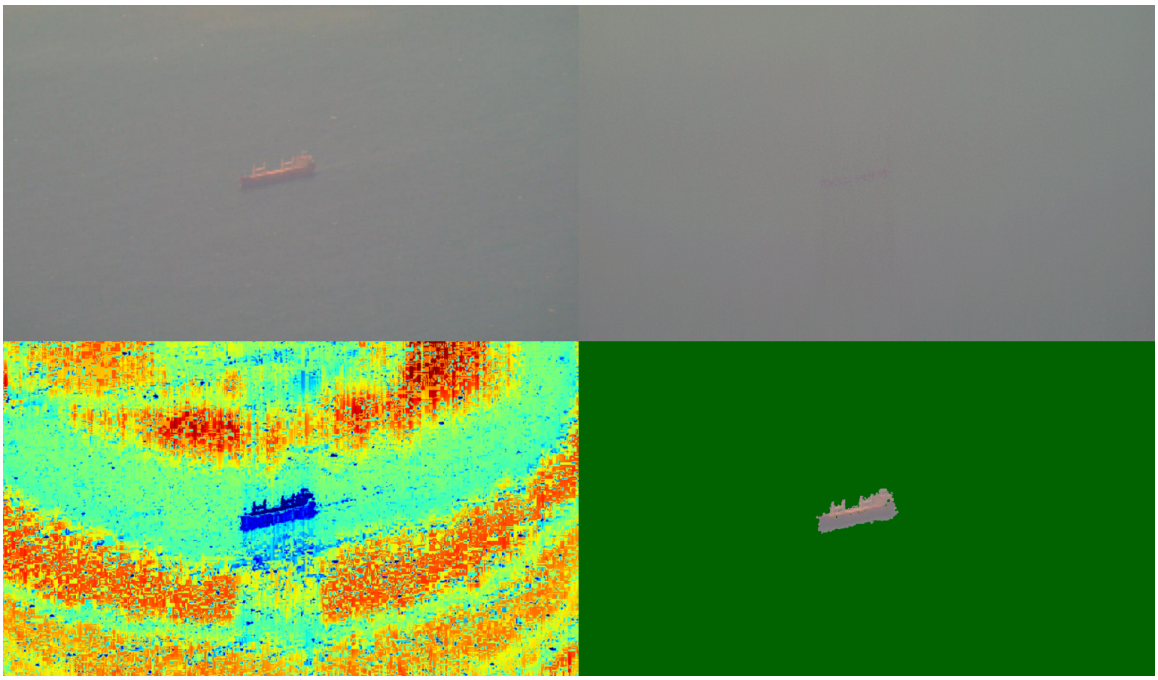

Figure 4.3: Sample input, re-generated ocean background, probabilistic map and final extraction result of the proposed algorithm

characteristics, as shown in Figure 4.3. First, the scene is commonly low-contrast, which tends to make traditional saliency detectors fail. In particular, there is no distinct texture appearing on an object, and objects sometimes fuse into the ocean background. Second, objects will be extremely small in the long-range scene, which is out of most saliency detectors' consideration. Finally, abnormal waves also meet the definition of a salient object, and thus employing general saliency detection methods would generate a large number of false alarms for a ship detection application. As the salient objects on the ocean vary, this work solves the problem by modeling the ocean background.

In the literature, most ship detection works by focusing on a top-down remote scene [70] [116] [78] [123] [104], where the size and shape of the ships are highly

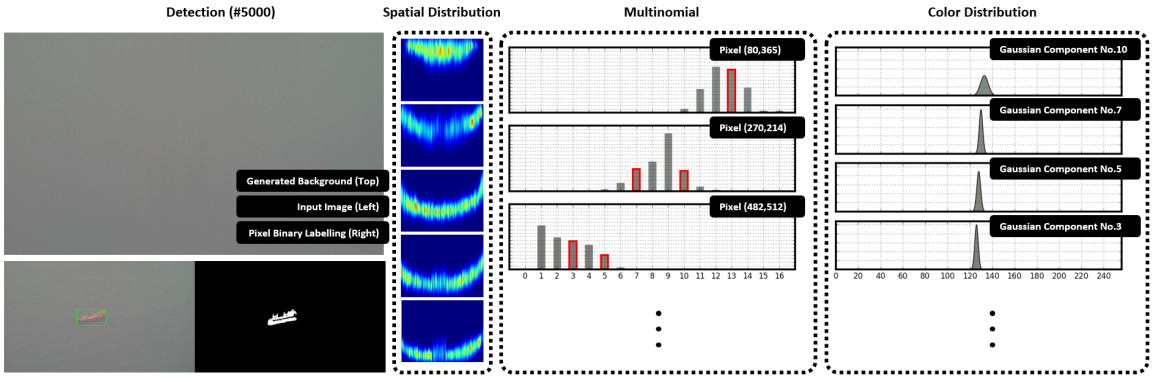

Figure 4.4: Algorithm representation

constrained, which simplifies the task. Many researchers formulate it as a binary classification problem. In [98], a model based on the HOG feature [26] and AdaBoost algorithm [110] is proposed to validate the ship candidates in a coarse-to-fine framework. [141] designs a classifier based on convolutional neural networks [105] and the singular value decompensation algorithm, which can adaptively learn the features from a spaceborne optical scenario. Some also make use of advanced image processing techniques. [122] design a novel hierarchical saliency filtering method to effectively detect ships in SAR images. In [107], the wavelet coefficient cue is used to improve detection accuracy. However, little of the literature on detecting ships reports work on the airborne surveillance scene as shown in Figure 4.3, where the object's appearance shows more variation and the background tends to be more complex.

In this work, an airborne ocean background modeling method is proposed to handle the task of extracting objects of interest. For the input image, each pixel is considered to be drawn from a Gaussian Mixture Model [38]. Rather than assuming independence between every pixel, this work asumes that all pixels share the same set of Gaussian components. This design is founded on the dynamic nature of the ocean background, where the color appearing on one background pixel could potentially

show up in any other background positions. However, the component weights vary with different pixel positions, which are controlled by a set of spatial distributions. The spatial distribution is designed based on the observation of ocean background from an airborne view, where pixels with similar colors tend to cluster together in the image plane and the whole background appears to be sutured together by a number of these clusters, as shown in Figure 4.4. Together, these distributions are formulated into a hierarchical model. Based on the model, the proposed work not only expresses the consistency of ocean appearance but also tolerates the color variation of its texture. To ensure that the model is adaptive to distinct scenes, an online learning method is designed to infer the parameters and the number of components simultaneously.

4.2 Algorithm

4.2.1 Representation

In this hierarchical model, the observed variable is pixel intensity x_m at position m, while the hidden variables include the indicator variable of this pixel z_m showing that x_m is generated by the k-th component of the Gaussian Mixture Model, component weight $\boldsymbol{\pi}_m$, parameter $\boldsymbol{\theta}_k$ of the k-th Gaussian distribution, and parameter $\boldsymbol{\phi}_k$, specifying a designed spatial distribution over the image plane to control the component weight for different image positions. Parameter M denotes the number of pixels in the image and K denotes the component count of the model. Since each Gaussian component corresponds to one spatial distribution, there are also K spatial distributions. Note that the algorithm's input is one single-channel image.

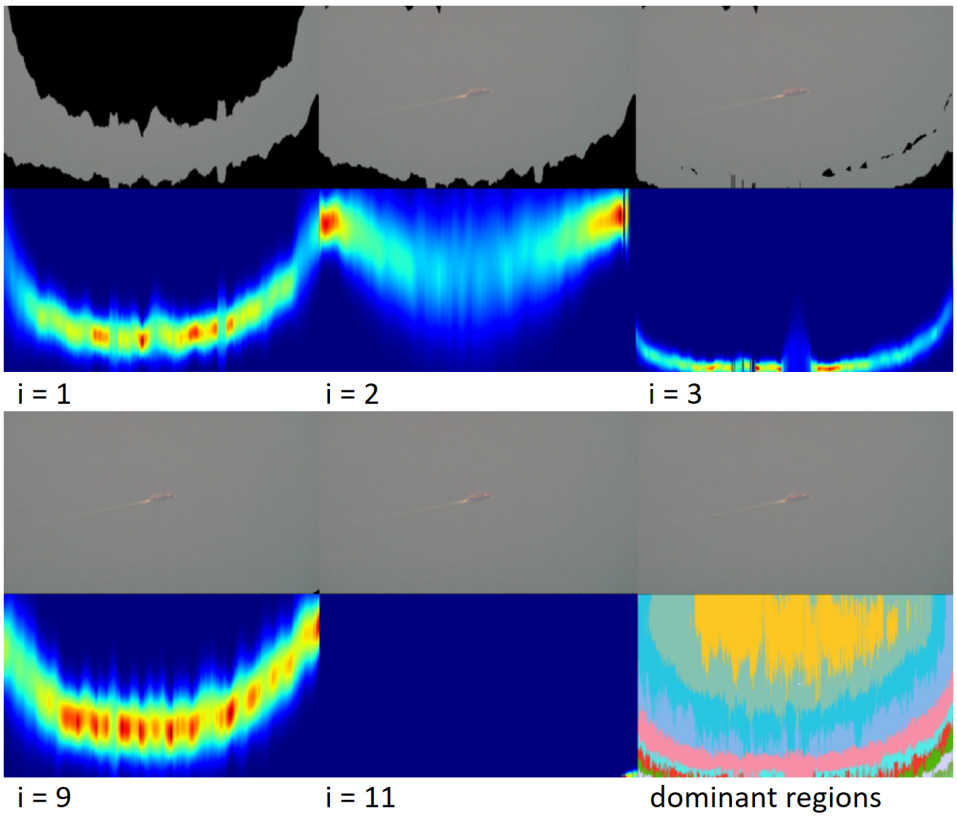

Figure 4.5: The global occupational regions and chosen spatial distributions in iterations of the algorithm inferring the number of components; the segmentation result of the dominant regions.

The dependency relationships between variables of this model can be expressed as follows:

$$\begin{aligned} \boldsymbol{\pi}_m|\{\boldsymbol{\phi}_k\}_{k=1}^K &\sim \mathcal{D}ir(p(m|\boldsymbol{\phi}_1),...,p(m|\boldsymbol{\phi}_K)) \\ z_m|\boldsymbol{\pi}_m &\sim \mathcal{M}ul(\boldsymbol{\pi}_m) \\ x_m|z_m,\{\boldsymbol{\theta}_k\}_{k=1}^K &\sim \mathcal{N}(\boldsymbol{\theta}_k). \end{aligned} \tag{4.3}$$

The generating process is that evidence x_m is drawn by one of the Gaussian distributions $\mathcal{N}(\boldsymbol{\theta}_k)$, indicated by z_m. Indicator variable z_m is randomly chosen by a

multinomial distribution $\mathcal{M}ul(\boldsymbol{\pi}_m)$, which essentially represents the weights of different Gaussian components for position m; $\boldsymbol{\pi}_m$ is generated by a Dirichlet distribution $\mathcal{D}ir(p(m|\boldsymbol{\phi}_1), ..., p(m|\boldsymbol{\phi}_K))$, where the k-th element of this parameter vector is the likelihood of position m of the k-th spatial distribution $p(m|\boldsymbol{\phi}_k)$. The formula of this spatial distribution is presented in Subsection 4.2.3.

4.2.2 Pixel Labeling

Given a learned model, the inference of likelihood that a pixel is generated by this model $p(x_m|\{\boldsymbol{\theta}_k\}_{k=1}^K, \{\boldsymbol{\phi}_k\}_{k=1}^K)$ is first presented.

Taking advantage of the conjugate relationship between Dirichlet and multinomial distributions, the density of indicator variable z_m can be calculated by Equation (4.4).

$$\begin{aligned} & p\left(z_m{=}k|\{\boldsymbol{\phi}_k\}_{k=1}^K\right) \\ &= \int p(z_m{=}k|\boldsymbol{\pi}_m) p(\boldsymbol{\pi}_m|\{\boldsymbol{\phi}_k\}_{k=1}^K) d\boldsymbol{\pi}^m \\ &= \frac{p(m|\boldsymbol{\phi}_k)}{\sum_{k'=1}^K p(m|\boldsymbol{\phi}_{k'})}, \end{aligned} \tag{4.4}$$

where $\boldsymbol{\pi}_m$ is integrated out and the probability of indicator z_m being assigned with component k is equal to the ratio that the probability of the k-th spatial distribution at position m is divided by the sum of all probabilities. As a result, a one-to-one correspondence is built between spatial distribution set $\{\boldsymbol{\phi}_k\}_{k=1}^K$ and Gaussian distribution set $\{\boldsymbol{\theta}_k\}_{k=1}^K$. A higher ratio of spatial distribution denotes a greater chance that the intensity at this position is drawn from the corresponding Gaussian component.

Given intensity x_m, the joint likelihood that it is drawn from the k-th Gaussian

component can be calculated as

$$\begin{aligned} & p(z_m{=}k, x_m|\{\boldsymbol{\theta}_k\}_{k=1}^K, \{\boldsymbol{\phi}_k\}_{k=1}^K) \\ & \quad = p(z_m{=}k|\{\boldsymbol{\phi}_k\}_{k=1}^K) p(x_m|z_m{=}k, \{\boldsymbol{\theta}_k\}_{k=1}^K) \\ & \quad = \frac{p(m|\boldsymbol{\phi}_k)}{\sum_{k'=1}^K p(m|\boldsymbol{\phi}_{k'})} \mathcal{N}(x_m|\mu_k, \sigma_k), \end{aligned} \tag{4.5}$$

where the second term is derived by the Markov property and Bayes' rule. The goal is to calculate the likelihood $p(x_m|\{\boldsymbol{\theta}_k\}_{k=1}^K, \{\boldsymbol{\delta}_k\}_{k=1}^K)$ that the evidence belongs to this model, which can be done by summing out the indicator variable z_m of Equation (4.5).

$$\begin{aligned} & p(x_m|\{\boldsymbol{\theta}_k\}_{k=1}^K, \{\boldsymbol{\phi}_k\}_{k=1}^K) \\ & \quad = \sum_{z_t^m=1}^K p(z_t^m, x_t^m|\{\boldsymbol{\theta}_k\}_{k=1}^K, \{\boldsymbol{\phi}_k\}_{k=1}^K). \end{aligned} \tag{4.6}$$

To identify any outliers, a threshold $\mathcal{P}_{th}$ is applied. If the above likelihood is higher, the pixel is labeled as ocean; otherwise, it is labeled as foreground objects.

4.2.3 Spatial Distributions

From the airborne view, an ocean background appears to be a mixture of multiple band-analogue regions of similar colors, as shown in the second block of Figure 4.4, although the region usually appears to have an irregular shape, caused by waves and imaging. To represent this spatial characteristic, a set of flexible spatial distributions are proposed.

Each spatial distribution is structured by a number of weighted one-dimensional

Gaussian distributions over different image columns. The distribution $p(m|\phi)$ can be expressed as

$$p(m|\boldsymbol{\phi}) = \sum_{j=1}^{J} w_j \delta_j(m) \mathcal{N}(m|\lambda_j, \rho_j), \tag{4.7}$$

where J is the number of columns in the image plane, w_j is the weight for the Gaussian distribution on the j-th column, and indicator function $\delta_j(m)=1$ if position m is in column j. The terms λ_j and ρ_j are the mean and variance of the Gaussian distribution, respectively, where different symbols are used for distinguishing the Gaussian component $\boldsymbol{\theta}=(\mu, \sigma)$ over intensity space.

Given a set of discrete image-position evidences $\{m\}$ of intensity x, the three kinds of parameters of this distribution $\phi = \{w_j, \lambda_j, \rho_j\}_{j=1}^{J}$ can be estimated. For each column, these vertical positions are recorded as $\boldsymbol{s}_j$. Here, a smoothing mechanism is employed, where the positions in nearby columns with distance smaller than ϵ are also accumulated into $\boldsymbol{s}_j$. Then, the weight w_j is calculated as the ratio of the evidence count to the total.

$$w_j = \frac{|\boldsymbol{s}_j|}{\sum_{j=1}^{J} |\boldsymbol{s}_j|}, \tag{4.8}$$

where $|\cdot|$ stands for the number of elements in this set. If number $|\boldsymbol{s}_j|$ is greater than zero, the column is marked as active. The estimator of mean λ_j is

$$\lambda_j = \frac{1}{|\boldsymbol{s}_j|} \sum_{\boldsymbol{s}_j} m, \tag{4.9}$$

where m here is a vertical coordinate in set $\boldsymbol{s}_j$. Similarly, the variance can be calculated as

$$\rho_j^2 = \frac{1}{|\boldsymbol{s}_j|} \sum_{\boldsymbol{s}_j} (m - \lambda_j)^2 . \tag{4.10}$$

4.2.4 Number of Components

For most unsupervised learning methods, the number of components K has to be pre-set, which requires previous knowledge of the data. Usually, this pre-setting is not optimal. In this work, an automatic inference method is present, as shown in Figure 4.5.

First, two concepts utilized in the method are introduced. A principle intensity x_k^p is defined as the intensity that spans a large enough region on the image plane and is closest to the mean of intensities appearing in this spanned region. For each principle intensity, a Gaussian component and a spatial distribution are created. The image in the upper left of Figure 4.5 shows the spatial distribution of principle intensity $x_k^p = 131$ and the image in the upper right of Figure 4.4 shows its Gaussian distribution. As a result, the task of inferring the number of components is converted to a task of determining a set of principle intensities. Another concept is occupational region $\boldsymbol{r}_x = \{m\}$, which can be derived from the spatial distribution. For each active column of the spatial distribution, the positions within two standard deviations are labeled as the occupational region. The image at the upper left of Figure 4.5 shows the occupational region $\boldsymbol{r}_{x_k^p}$ of principle intensity $x_k^p = 131$. There is also a global occupational region $\boldsymbol{r}$, which is defined as the union set of all occupational regions of principle intensities.

This algorithm starts from the intensity appearing the most frequently in the image, which is labeled as the first principle intensity x_1^p. Making use of its positions, spatial distribution $\boldsymbol{\phi}_1$ can be estimated based on Equation (4.8)-(4.10). As a result, occupational region $\boldsymbol{r}_{x_1^p}$ can be obtained. This region is then updated into the global occupational region, denoted as $\boldsymbol{r}^1 = \boldsymbol{r}_{x_1^p}$, where the superscript indicates the iteration

count. Then, iterate the following steps: 1) in iteration i, if there is still intensity without being included in the principle set, search through all; 2) for each x, estimate its occupational region $\boldsymbol{r}_x$; 3) combine this region with $\boldsymbol{r}^{i-1}$ to form a candidate current global occupational region $\boldsymbol{r}_x^i$; 4) calculate the ranking score for x using the following equation:

$$score\,(x) = \frac{|\boldsymbol{r}_x^i - \boldsymbol{r}^{i-1}|}{unique\,(\boldsymbol{r}_x^i - \boldsymbol{r}^{i-1})}, \tag{4.11}$$

which represents the number of pixels per unique intensity of the increased global region $\boldsymbol{r}_x^i - \boldsymbol{r}^{i-1}$. $unique(\cdot)$ denotes the number of unique elements in a set; 5) label the intensity with the highest score as the new principle intensity and update the new global occupational region $\boldsymbol{r}^i$; and 6) check the ratio of $\boldsymbol{r}^i$ to the image size. If it is greater than an occupational threshold $\mathcal{O}_{th}$, the algorithm terminates; otherwise, go back to step 1). Figure 4.5 shows the global occupational regions $\boldsymbol{r}^i$ and the spatial distributions of selected principle intensities $\boldsymbol{\phi}_k$ in the sample iterations.

4.2.5 Gaussian Distributions

The algorithm discussed above simultaneously infers the number of components and each component's spatial distribution, which leaves the set of Gaussian distributions to be estimated.

The mean μ_k of the k-th Gaussian component is set as the corresponding value of the principle intensity, as

$$\mu_k = x_k^p. \tag{4.12}$$

The reason behind this estimator is that the essential goal is to accurately detect the outliers in a potential low-contrast scene. A labeled principle intensity in the

aforementioned algorithm ensures that it covers a large enough area with consistent color. This feature is sufficient for outlier detection. An complicated estimation process may introduce more noise from the unclean data.

The mission tolerating the variation of the ocean background is left to Gaussian variance σ_k^2. Before presenting the estimator, the dominant region $\boldsymbol{r}_k^d$ of each component k is introduced. The dominant region of component k is a set of positions where probability $p(m|\boldsymbol{\phi}_k)$ is greater than other spatial distributions. As shown in the image at the lower right of Figure 4.5, the global occupational region $\boldsymbol{r}$ is exclusively segmented by a set of dominant regions $\{\boldsymbol{r}_k^d\}$. Note that the difference between dominant region $\boldsymbol{r}_k^d$ and occupational region $\boldsymbol{r}_{x_k^p}$ is that occupational regions may overlap each other, while dominant regions do not. In dominant region $\boldsymbol{r}_k^d$, the pixel intensities appearing in the eight-neighborhood of a principle intensity x_k^p are denoted as $\boldsymbol{r}_k^{d,+}$. This pixel set is used to estimate the Gaussian variance parameter of the corresponding principle intensity.

$$\sigma_k^2 = \frac{1}{|\boldsymbol{r}_k^{d,+}|} \sum_{\boldsymbol{r}_k^{d,+}} (x_m - \mu_k)^2 \tag{4.13}$$

Specifically, on the one hand, the variance would not be too large to cover the outlier intensity and, on the other hand, it could tolerate the local change of ocean background.

4.3 Experiment

After thresholding, the original binary mask might still contain ocean pixels: some blobs may be caused by an anomaly wave. A blob-size threshold is applied, in which

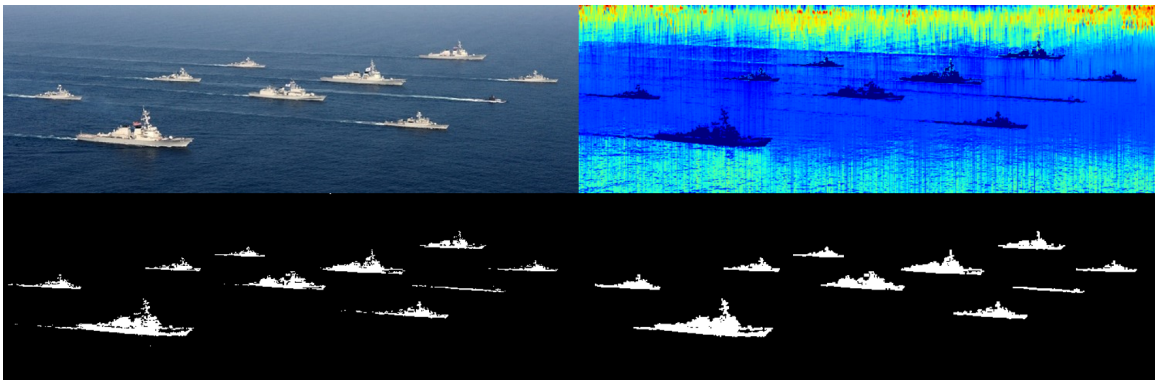

Figure 4.6: A blob size threshold is applied to the original mask obtained by thresholding the probabilistic map.

only a blob larger than this threshold will be treated as an object. In Figure 4.6, the original mask is shown on the left, while the processed one is on the right. Note that, in real-application, this threshold does not need to be manually set. It can be inferred by the parameters of the sensor and the movement status of the aircraft.

In this chapter, the experiments are conducted based on four datasets, namely *small*, *big*, *thermal*, and *web*. The first three are ocean surveillance videos captured by aircraft, as shown in Figure 4.12, Figure 4.13 and Figure 4.14. In this test, every frame of each video is treated as an individual image without utilizing any prior information of the corresponding sequence. The last dataset *web* is a set of images collected from the Internet, which is used to test various complex scenes. These images are grouped into four classes, *single object scene*, *multiple objects scene*, *rescue scene*, and *accident scene*. Together, there are more than $30,000$ frames tested in this experiment. The system is implemented in C/C++, which runs on a Core i7 CPU with 3.5 GHz and 12 GB memory computer. For a resolution 320×240 image, the system needs 0.047 seconds based on single-core implementation.

The proposed algorithm has a few hyper-parameters to set before being applied.

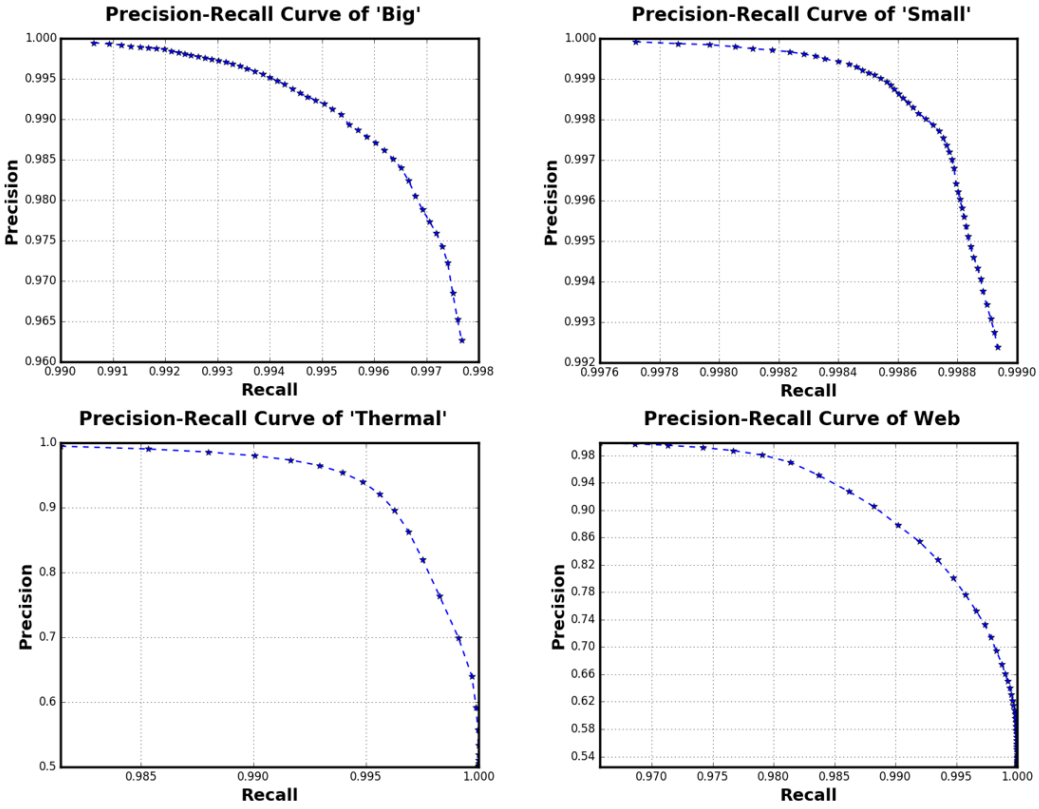

Figure 4.7: The Precision-Recall curves of proposed algorithm on four datasets

The first is the occupation threshold in the algorithm for determining principle intensities, which is set as $\mathcal{O}_{th} = 0.95$. The overlap width for estimating the Gaussian parameter of a single column of spatial distributions is set as $\epsilon = 9$. Finally, for the probabilistic threshold, the default setting is $\mathcal{P}_{th} = 0.01$. All three hyper-parameters are fixed for all tested images.

To quantitatively evaluate the algorithm's performance, ground truth binary masks are manually labeled. For the three video datasets, 300 frames are randomly chosen from each. A total of 200 images are labeled in the *web* dataset. In this work, two metrics are employed, *recall* and *precision*. *Precision* measures the rate of true

Figure 4.8: Sample results for *accident scene* of *web* dataset

positives tp out of the total detections, which can be calculated as

$$precision = \frac{tp}{tp + fp}, \tag{4.14}$$

where fp indicates the number of false positives. *Recall* is defined as

$$recall = \frac{tp}{tp + fn}, \tag{4.15}$$

where fn is the false negative count.

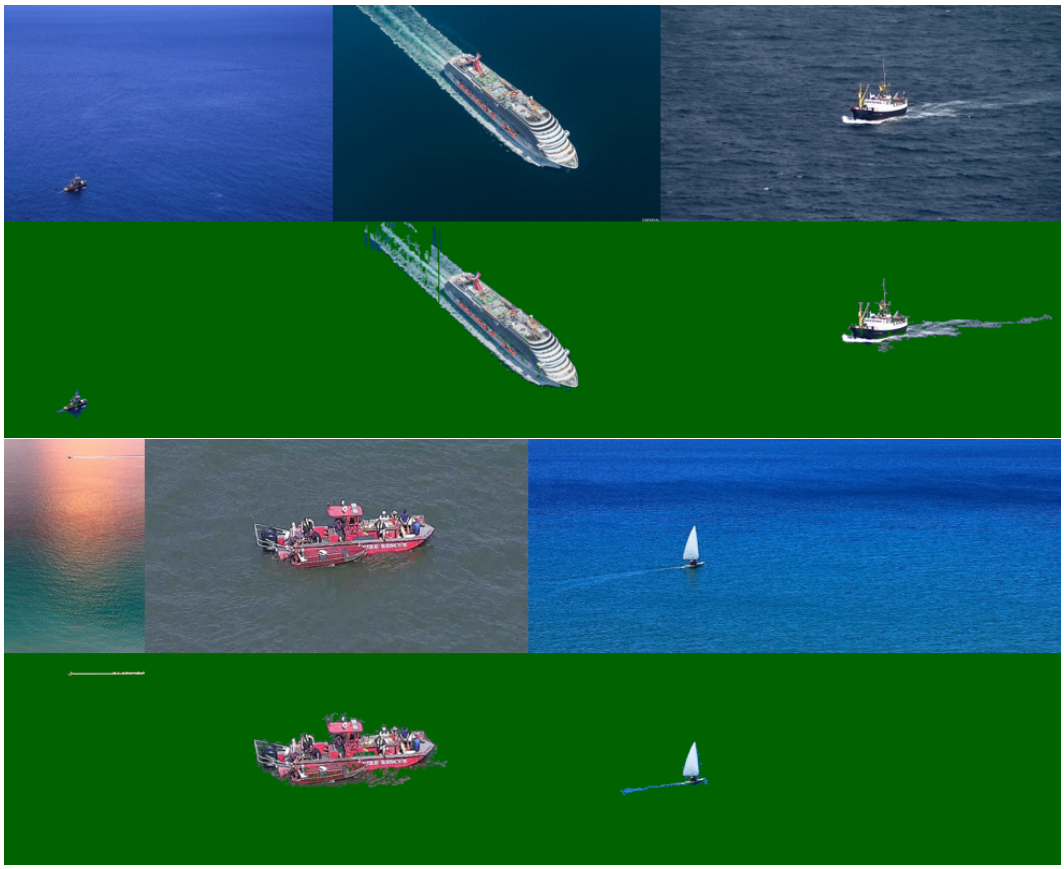

Figure 4.9: Sample results for *single object scene* of *web* dataset

4.3.1 Performance Analysis

Figure 4.8 to Figure 4.14 show sample results of the proposed algorithm on the four datasets. Figure 4.7 shows the corresponding precision-recall curves. Note that, in these curves, the blob-size threshold is not applied for purely testing the algorithm's capability of modeling ocean background.

In dataset *small*, the object size is smaller than 500 pixels and the scene is low contrast. Specifically, each object intensity massively occurs in the ocean background. In addition, there is no obvious texture or gradient around the ship position. Together,

Figure 4.10: Sample results for *rescue scene* of *web* dataset

these factors make the classifier-based detector and saliency-based detector inapplicable. The background for the *big* dataset keeps changing throughout the sequence. As shown in Figure 4.13, the scene appears to be low contrast at the beginning. As the vessel gets closer, more and more texture is revealed. At the end of the video, the scene contains a large number of waves. Dataset *thermal* is employed to test this proposed method on a thermal sensor. In this scene, the signal-to-noise ratio is low. Although there is prior knowledge that an object tends to be lighter based on the principle of thermal imaging, this method does not exploit this information, since it is designed to handle various scenes, not just a thermal scenario. From the precision-recall curves and sample results, this algorithm shows accuracy and robustness on all three datasets.

Figure 4.11: Sample results for *multiple objects scene* of *web* dataset

The *web* dataset was collected from the Internet, which has better image quality than the three previously tested scenes. These Internet scenes are more complex, and can be grouped into four different subsets based on the objects present. For the first class, *single object scene*, the proposed algorithm extracts the objects properly. It can be observed that object size does not affect this method, which works well from a small object of less than 50 pixels to a large one that occupies almost one-third of the image. It is also observed that the method can tolerate a sudden change in the ocean's color, which can be caused by sunlight or a change in depth. Of the four classes, our algorithm performs the best on the *multiple objects scene*, as shown in Figure 4.11. Even when the sizes of objects in the same image differ significantly, distinct kinds of objects such as a boat or human appear together, and the angles of the airborne cameras vary across the image set, the method works well. The third

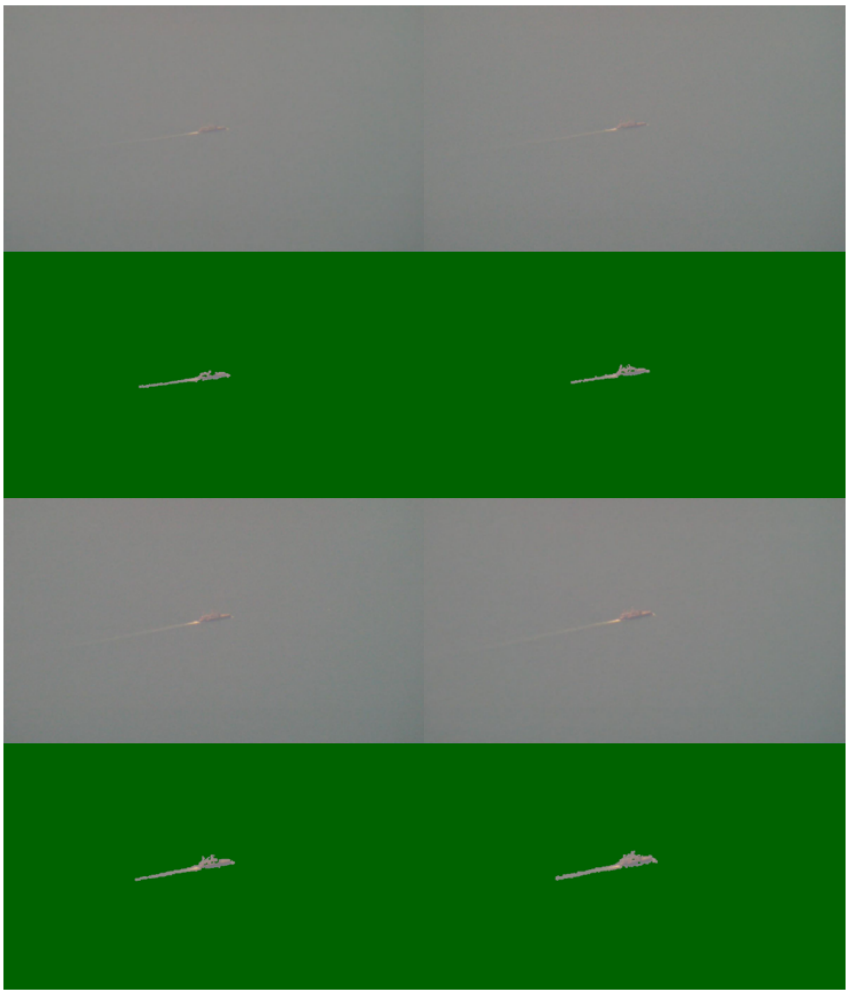

Figure 4.12: Sample results for dataset *small*

group is *rescue scene*. In this set, the present objects are a human, lifeboat, overturned ship, and anomalous objects of interest. Detection of these types of objects is the most important task of an airborne surveillance platform. They usually show as irregular shapes and are hard to classify. However, the proposed method reports good performance on extraction. Detecting accidents is also a critical task for ocean surveillance. The last set *accident scene* includes images of actual accidents at sea.

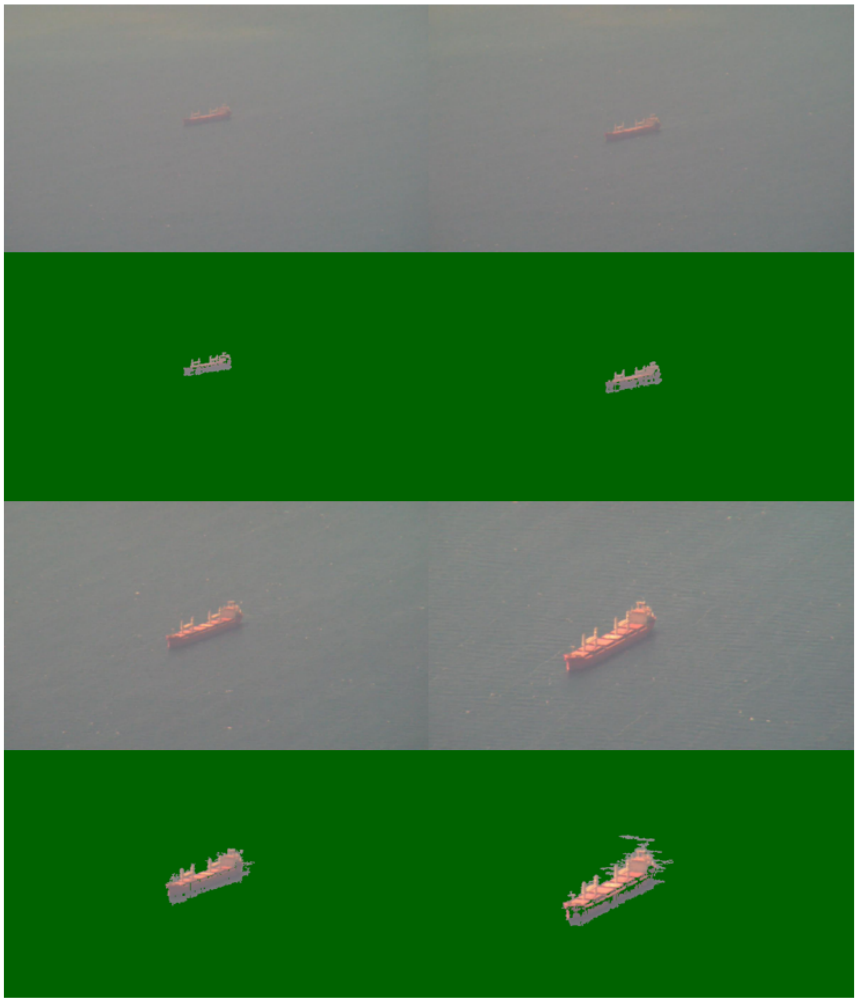

Figure 4.13: Sample results for dataset *big*

Usually, there is smoke in the image, along with the object. Surprisingly, this algorithm handles these scenes well. The author note that, for accident detection, the proposed system performs on better for long-range scenes than for close-range scenes. The reason is that the smoke occupies a extremely large portion of the close-range scene, which makes the algorithm treat it as the principle intensity. In general, the proposed algorithm models the ocean background well in this *web* dataset, which is consistent with the precision-recall curve of Figure 4.7.

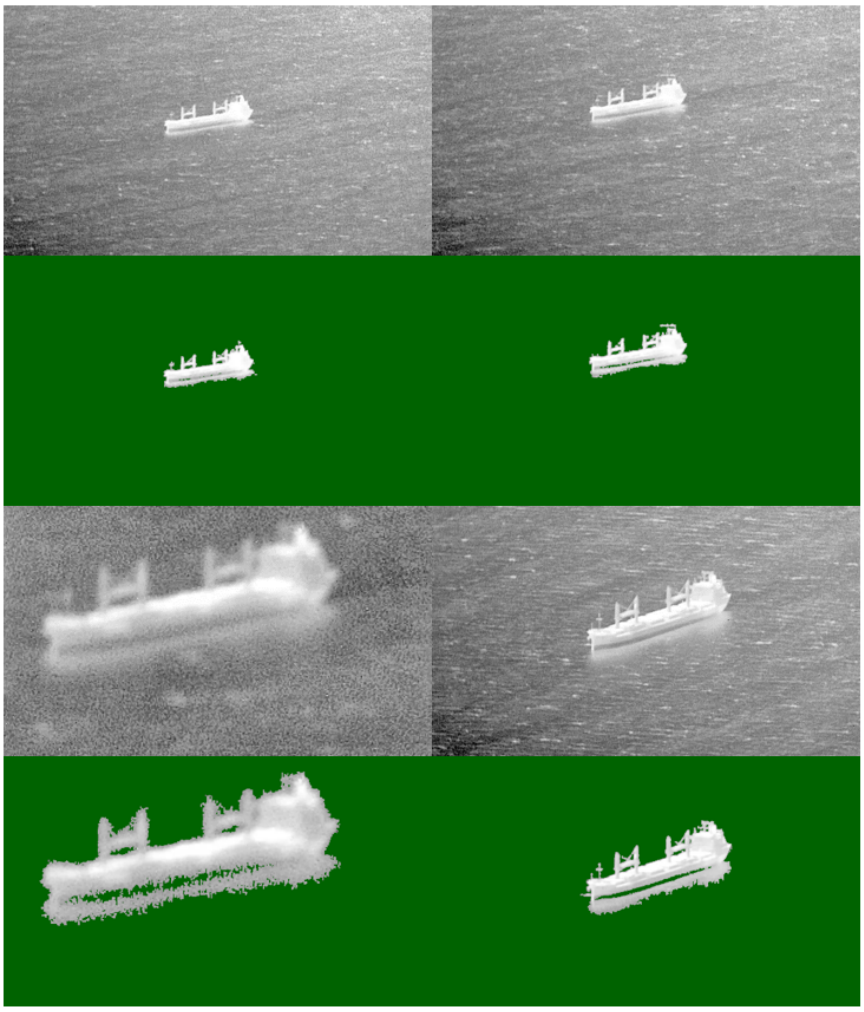

Figure 4.14: Sample results for dataset *thermal*

4.3.2 Module Analysis

To further test the proposed method's capability of modeling ocean background, Figure 4.15 shows a set of reconstructed ocean background based on learned models. Since the proposed model is a generative model, the reconstructed background can be drawn from it. The figure shows that each ocean background with an object absent is successfully reconstructed, which also represents the original ocean texture's

Figure 4.15: Results of background reconstruction

characteristics. For instance, in the first pair, the light red color appearing in the middle of the image is fully encoded into this new background. But for local details, two images are shown differently, which is caused by the random sampling. From the model view, the original background only represents one possible occurrence.

4.4 Conclusion

In this chapter, an ocean background modeling algorithm based on a single image is proposed. A statistical hierarchical model is utilized to represent the ocean background, making use of Gaussian Mixture Model and designed spatial distributions,

which together formulate the objects of interest extraction process as an outlier detection task. To infer the model's parameters, an online learning method is employed. This approach is able to learn the number of components and component parameters simultaneously. An intensive experiment demonstrates the accuracy and robustness of this algorithm.

Chapter 5

Conclusion

5.1 Research Summary

In this thesis, a simple but effective framework representing a grid of image-domain label-distributions is proposed to universally improve the performance of a number of computer vision problems. Based on the framework, a temporal or spatial consistency relationship between appearance components can be kept for different vision tasks. To demonstrate this usefulness, it has been successfully applied to three particular applications.

For object tracking task, it was cast as a prior regularized semi-supervised learning problem by combining the framework with the Random Forest classifier. In this algorithm, a novel patch-based-grid target representation was designed taking advantage of discriminative appearance model and generative structure model. The appearance model attempted to classify the foreground and background using not only the historical data but also the current frame data. The grid of label-distributions

was formulated as a regularizer to constrain the learning process. A heuristic multi-objective optimization method was proposed to find the solution, which ensured object appearance and object structure collaboratively optimized. The experiments on standard sequences showed that this proposed algorithm outperformed existing approaches that especially handled the effects caused by pose, rotation, illumination, blur, abrupt motion, occlusion and background clutter.

For background subtraction task, it was firstly formulated as a probabilistic topic model by fusing the framework with the Gaussian mixture model. Taking advantage of a newly designed on-line learning algorithm, the topics and topic proportions of the topic model were efficiently updated. This model kept a set of stable Gaussian topics representing the real background objects rather than the instant evidence and flexible topic proportions to reflect the movement of background objects. As a result, the method suitably modelled the motion of dynamic backgrounds and outperformed state-of-the-art methods on multiple challenging datasets.

For ship detection task, a novel airborne ocean background modelling method based on single input image was proposed by injecting a hyper-distribution dominating the spatial information into the framework. A statistical hierarchical model was utilized to represent the ocean background, making use of Gaussian Mixture Model and the designed spatial distribution, which together formulated the objects of interest extraction process as an outlier detection task. To infer the parameters, an on-line learning method was employed. This approach was able to learn the number of appearance components and component parameters simultaneously. An intensive experiment demonstrated the accuracy and robustness of this algorithm.

5.2 Future Work

For future research, the proposed framework can be studied to be applied on other computer vision tasks. Generally speaking, this framework appears to be applicable on most video analysis problems. For example, multi-target tracking task can make use of it, as the grid presentation provides additional measurement to differing distinct targets in the scene. The single object tracking task can also be extended for extracting segmentation result by further exploring the grid presentation. This framework can also be utilized for some single image processing problem. For instance, it can be applied to road surface extraction for automatic driving by following the same methodology of proposed ship detection algorithm. Furthermore, using more advanced appearance model can potentially improve the ship detection method.

Bibliography

[1] (2017). *A Hybrid of Local and Global Saliencies for Detecting Image Salient Region and Appearance*, volume 47.

[2] Adam, A., Rivlin, E., and Shimshoni, I. (2006). Robust fragments-based tracking using the integral histogram. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805.

[3] Allebosch, G., Deboeverie, F., Veelaert, P., and Philips, W. (2015). Efic: edge based foreground background segmentation and interior classification for dynamic camera viewpoints. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 130–141. Springer.

[4] Allili, M. S., Bouguila, N., and Ziou, D. (2007). A robust video foreground segmentation by using generalized Gaussian mixture modeling. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 503–509.

[5] Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Comput.*, **9**(7), 1545–1588.

[6] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, **50**(2), 174–188.

[7] Avidan, S. (2004). Support vector tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(8), 1064–1072.

[8] Avidan, S. (2007). Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**(2), 261–271.

[9] Babenko, B., Yang, M.-H., and Belongie, S. (2011). Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **33**(8), 1619–1632.

[10] Babu, R. V., Parate, P., and niruddha Acharya K. (2015). Robust tracking with interest points: A sparse representation approach. *Image and Vision Computing*, **33**, 44 – 56.

[11] Balan, A. and Black, M. (2006). An adaptive appearance model approach for model-based articulated object tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 758–765.

[12] Barnich, O. and Droogenbroeck, M. V. (2009). Vibe: A powerful random technique to estimate the background in video sequences. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 945–948.

[13] Benezeth, Y., Jodoin, P.-M., Emile, B., Laurent, H., and Rosenberger, C. (2010). Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, **19**(3), 033003–033003–12.

[14] Bouwmans, T. (2011-09-01T00:00:00). Recent advanced statistical background modeling for foreground detection - a systematic survey. *Recent Patents on Computer Science*, **4**(3), 147–176.

[15] Bouwmans, T. (2014). Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, **1112**, 31 – 66.

[16] Breiman, L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32.

[17] Carminati, L. and Benois-Pineau, J. (2005). Gaussian mixture classification for moving object detection in video surveillance environment. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–113–16.

[18] Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-supervised Learning*. The MIT Press.

[19] Chen, Y., Wang, J., and Lu, H. (2015). Learning sharable models for robust background subtraction. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[20] Cheng, L. and Gong, M. (2009). Realtime background subtraction from dynamic scenes. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2066–2073.

[21] Collins, R., Liu, Y., and Leordeanu, M. (2005). Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **27**(10), 1631–1643.

[22] Corbane, C., Marre, F., and Petit, M. (2008). Using spot-5 hrg data in panchromatic mode for operational detection of small ships in tropical area. *Sensors*, **8**(5), 2959–2973.

[23] Corduneanu, A. and Jaakkola, T. (2003). On information regularization. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, pages 151–158, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[24] Criminisi, A. and Shotton, J. (2013). *Decision forests for computer vision and medical image analysis*. Advances in Computer Vision and Pattern Recognition. Springer, Dordrecht.

[25] Crisp, D. (2004). The state-of-the-art in ship detection in synthetic aperture radar imagery.

[26] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1.

[27] Daum, F. (2005). Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, **20**(8), 57–69.

[28] de Jong, A. N. (1993). Ship infrared detection/vulnerability.

[29] deSilva, C. J. S., Lee, G., and Johnson, R. (1995). All-aspect ship recognition in infrared images. In *Proceedings Electronic Technology Directions to the Year 2000*, pages 194–198.

[30] Ding, J., Li, M., Huang, K., and Tan, T. (2011). *Modeling Complex Scenes for Accurate Moving Objects Segmentation*, pages 82–94. Springer Berlin Heidelberg, Berlin, Heidelberg.

[31] Du, S. and Chen, S. (2014). Salient object detection via random forest. *Signal Processing Letters, IEEE*, **21**(1), 51–54.

[32] Eldhuset, K. (1996). An automatic ship and ship wake detection system for spaceborne sar images in coastal regions. *IEEE Transactions on Geoscience and Remote Sensing*, **34**(4), 1010–1019.

[33] Elgammal, A., Harwood, D., and Davis, L. (2000). *Non-parametric Model for Background Subtraction*, pages 751–767. Springer Berlin Heidelberg, Berlin, Heidelberg.

[34] Elhabian, S. Y., El-Sayed, K. M., and Ahmed, S. H. (2008-01-01T00:00:00). Moving object detection in spatial domain using background removal techniques - state-of-the-art. *Recent Patents on Computer Science*, **1**(1), 32–54.

[35] Everingham, M., Gool, L., Williams, C., and Zisserman, A. (2005). Pascal visual object classes challenge results. *Available from www. pascal-network. org*.

[36] Fan, W. and Bouguila, N. (2012). Online variational learning of finite Dirichlet mixture models. *Evolving Systems*, **3**(3), 153–165.

[37] Fang, Y., Chen, Z., Lin, W., and Lin, C. W. (2012). Saliency detection in the compressed domain for adaptive image retargeting. *IEEE Transactions on Image Processing*, **21**(9), 3888–3901.

[38] Figueiredo, M. A. T. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(3), 381–396.

[39] Grabner, H., Grabner, M., and Bischof, H. (2006). Real-time tracking via on-line boosting. In *Proc. BMVC*, pages 6.1–6.10. doi:10.5244/C.20.6.

[40] Grabner, H., Leistner, C., and Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision ECCV 2008*, volume 5302 of *Lecture Notes in Computer Science*, pages 234–247. Springer Berlin Heidelberg.

[41] Grandvalet, Y. and Bengio, Y. (2005). Semi-supervised learning by entropy minimization.

[42] Guo, L. and h. Du, M. (2012). Student's t-distribution mixture background model for efficient object detection. In *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on*, pages 410–414.

[43] Haines, T. S. F. and Xiang, T. (2014). Background Subtraction with Dirichlet Process Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(4), 670–683.

[44] Han, B. and Lin, X. (2005). Update the GMMs via adaptive Kalman filtering. volume 5960, pages 59604F–59604F–10.

[45] Haritaoglu, I., Harwood, D., and Davis, L. S. (2000). W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 809–830.

[46] He, Y., Wang, D., and Zhu, M. (2011). Background subtraction based on non-parametric Bayesian estimation. volume 8009, pages 80090G–80090G–5.

[47] Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**(3), 583–596.

[48] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, **313**, 504–507.

[49] Hofmann, M., Tiefenbacher, P., and Rigoll, G. (2012). Background segmentation with feedback: The pixel-based adaptive segmenter. In *2012 IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition Workshops, pages 38–43.

[50] Jaakkola, M. S. T. and Szummer, M. (2002). Partially labeled classification with markov random walks. *Advances in Neural Information Processing Systems (NIPS)*, **14**, 945–952.

[51] Jia, X., Lu, H., and Yang, M. H. (2012). Visual tracking via adaptive structural local sparse appearance model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1822–1829.

[52] Kalal, Z., Matas, J., and Mikolajczyk, K. (2010). P-n learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56.

[53] Kalal, Z., Mikolajczyk, K., and Matas, J. (2012). Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **34**(7), 1409–1422.

[54] Kim, H., Sakamoto, R., Kitahara, I., Toriyama, T., and Kogure, K. (2007). Robust silhouette extraction technique using background subtraction. In *10th Meeting on Image Recognition and Understand, MIRU*.

[55] Kwon, J. and Lee, K. M. (2010). Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276.

[56] Kwon, J. and Lee, K. M. (2013). Highly nonrigid object tracking via patch-based

dynamic appearance modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(10), 2427–2441.

[57] Lai, A. H. S. and Yung, N. H. C. (1998). A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence. In *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, volume 4, pages 241–244 vol.4.

[58] Lee, D.-S. (2002). Improved adaptive mixture learning for robust video background modeling. In *MVA*, pages 443–446. Citeseer.

[59] Lee, D.-S. (2005). Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(5), 827–832.

[60] Leibe, B., Schindler, K., Cornelis, N., and Van Gool, L. (2008). Coupled object detection and tracking from static cameras and moving vehicles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **30**(10), 1683–1698.

[61] Leistner, C., Saffari, A., Santner, J., and Bischof, H. (2009). Semi-supervised random forests. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 506–513.

[62] Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **28**(9), 1465–1479.

[63] Li, A., Lin, M., Wu, Y., Yang, M., and Yan, S. (2015a). Nus-pro: A new visual

tracking challenge. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **PP**(99), 1–1.

[64] Li, H., Li, Y., and Porikli, F. (2016). Deeptrack: Learning discriminative feature representations online for robust visual tracking. *IEEE Transactions on Image Processing*, **25**(4), 1834–1848.

[65] Li, L., Huang, W., Gu, I. Y. H., and Tian, Q. (2003). Foreground object detection from videos containing complex background. In *Proceedings of the Eleventh ACM International Conference on Multimedia*, MULTIMEDIA '03, pages 2–10, New York, NY, USA. ACM.

[66] Li, Y., Zhu, J., and Hoi, S. C. (2015b). Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[67] Lin, H. H., Liu, T. L., and Chuang, J. H. (2009). Learning a scene background model via classification. *IEEE Transactions on Signal Processing*, **57**(5), 1641–1654.

[68] Lin, L., Xu, Y., Liang, X., and Lai, J. (2014). Complex background subtraction by pursuing dynamic spatio-temporal models. *IEEE Transactions on Image Processing*, **23**(7), 3191–3202.

[69] Lindstrm, J., Lindgren, F., strm, K., Holst, J., and Holst, U. (2006). Background and foreground modeling using an online EM algorithm. In G. Jones, editor, *IEEE*

International Workshop on Visual Surveillance, volume VS2006, pages 9–16. Faculty of Computing, Information Systems and mathematics, Kingston University, Surrey, UK.

[70] Liu, G., Zhang, Y., Zheng, X., Sun, X., Fu, K., and Wang, H. (2014). A new method on inshore ship detection in high-resolution satellite images using shape and context information. *IEEE Geoscience and Remote Sensing Letters*, **11**(3), 617–621.

[71] Liu, X. and Yu, T. (2007). Gradient feature selection for online boosting. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8.

[72] Liu, Z., Chen, W., Huang, K., and Tan, T. (2008). A Probabilistic Framework Based on KDE-GMM Hybrid Model for Moving Object Segmentation in Dynamic Scenes. In *The Eighth International Workshop on Visual Surveillance - VS2008*, Marseille, France. Graeme Jones and Tieniu Tan and Steve Maybank and Dimitrios Makris.

[73] Liu, Z., Huang, K., and Tan, T. (2012). Foreground object detection using top-down information based on EM framework. *IEEE Transactions on Image Processing*, **21**(9), 4204–4217.

[74] Lure, F. Y. M. and Rau, Y.-C. (1994). Detection of ship tracks in avhrr cloud imagery with neural networks. In *Geoscience and Remote Sensing Symposium, 1994. IGARSS '94. Surface and Atmospheric Remote Sensing: Technologies, Data Analysis and Interpretation., International*, volume 3, pages 1401–1403 vol.3.

[75] Maddalena, L. and Petrosino, A. (2008). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, **17**(7), 1168–1177.

[76] Mahadevan, V. and Vasconcelos, N. (2010). Spatiotemporal saliency in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(1), 171–177.

[77] Mann, G. S. and McCallum, A. (2007). Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the 24th International Conference on Machine Learning*, pages 593–600.

[78] Marino, A. (2013). A notch filter for ship detection with polarimetric sar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **6**(3), 1219–1232.

[79] McFarlane, N. J. B. and Schofield, C. P. (1995). Segmentation and tracking of piglets in images. *Machine Vision and Applications*, **8**(3), 187–193.

[80] McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., and Wechsler, H. (2000). Tracking groups of people. *Computer Vision and Image Understanding*, **80**(1), 42 – 56.

[81] Mei, X. and Ling, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **33**(11), 2259–2272.

[82] Montgomery, D. C., Peck, E. A., and Vining, G. G. (2007). *Introduction to Linear*

Regression Analysis, Solutions Manual (Wiley Series in Probability and Statistics). Wiley-Interscience.

[83] Morellas, V., Pavlidis, I., and Tsiamyrtzis, P. (????). Deter: Detection of events for threat evaluation and recognition. *Machine Vision and Applications*, **15**(1), 29–45.

[84] Mukherjee, D. and JonathanWu, Q. (2012). Real-time video segmentation using student's t-mixture model. *Procedia Computer Science*, **10**, 153 – 160.

[85] Nejhum, S. M. S., Ho, J., and Yang, M. H. (2008). Visual tracking with histograms and articulating blocks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[86] Ng, K. W., Tian, G.-L., and Tang, M.-L. (2011). *Dirichlet and related distributions: Theory, methods and applications*. John Wiley & Sons.

[87] Payet, N. and Todorovic, S. (2013). Hough forest random field for object recognition and segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(5), 1066–1079.

[88] Pedersoli, M., Timofte, R., Tuytelaars, T., and Van Gool, L. (2015). An elastic deformation field model for object detection and tracking. *International Journal of Computer Vision*, **111**(2), 137–152.

[89] Piccardi, M. (2004). Background subtraction techniques: A review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104 vol.4.

[90] Pnevmatikakis, A. and Polymenakos, L. (2007). *2D Person Tracking Using Kalman Filtering and Adaptive Background Learning in a Feedback Loop*, pages 151–160. Springer Berlin Heidelberg, Berlin, Heidelberg.

[91] Porikli, F. and Tuzel, O. (2003). Human body tracking by adaptive background models and mean-shift analysis. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–9. Citeseer.

[92] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed gibbs sampling for latent Dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 569–577, New York, NY, USA. ACM.

[93] Ramrez-Alonso, G. and Chacn-Murgua, M. I. (2016). Auto-adaptive parallel {SOM} architecture with a modular analysis for dynamic object segmentation in videos. *Neurocomputing*, **175, Part B**, 990 – 1000.

[94] Ross, D., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, **77**(1-3), 125–141.

[95] Sajid, H. and Cheung, S. C. S. (2015). Background subtraction for static amp; moving camera. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4530–4534.

[96] Sevilla-Lara, L. and Learned-Miller, E. (2012). Distribution fields for tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1910–1917.

[97] Sheikh, Y. and Shah, M. (2005). Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(11), 1778–1792.

[98] Shi, Z., Yu, X., Jiang, Z., and Li, B. (2014). Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature. *IEEE Transactions on Geoscience and Remote Sensing*, **52**(8), 4511–4523.

[99] Shotton, J., Johnson, M., and Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[100] St-Charles, P. L., Bilodeau, G. A., and Bergevin, R. (2015). Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, **24**(1), 359–373.

[101] St-Charles, P. L., Bilodeau, G. A., and Bergevin, R. (2016). Universal background subtraction using word consensus models. *IEEE Transactions on Image Processing*, **25**, 4768–4781.

[102] Stalder, S., Grabner, H., and Gool, L. V. (2009). Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1409–1416.

[103] Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, page 252 Vol. 2.

[104] Suzuki, S., Mitsukura, Y., and Furuya, T. (2014). Ship detection based on spatio-temporal features. In *2014 10th France-Japan/ 8th Europe-Asia Congress on Mecatronics (MECATRONICS2014- Tokyo)*, pages 93–98.

[105] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

[106] Tan, R., Huo, H., Qian, J., and Fang, T. (2006). *Traffic Video Segmentation Using Adaptive-K Gaussian Mixture Model*, pages 125–134. Springer Berlin Heidelberg, Berlin, Heidelberg.

[107] Tang, J., Deng, C., Huang, G. B., and Zhao, B. (2015). Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. *IEEE Transactions on Geoscience and Remote Sensing*, **53**(3), 1174–1185.

[108] Teh, Y. W., Kurihara, K., and Welling, M. (2008). Collapsed variational inference for HDP. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1481–1488. Curran Associates, Inc.

[109] Timoftea, R., Kwona, J., and Goola, L. V. (2016). Picaso: Pixel correspondences and soft match selection for real-time tracking. *Computer Vision and Image Understanding*.

[110] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference*

on Computer Vision and Pattern Recognition. CVPR 2001, volume 1, pages I–511–I–518 vol.1.

[111] Vojíř, T. and Matas, J. (2014). *The Enhanced Flock of Trackers*, pages 113–136. Springer Berlin Heidelberg, Berlin, Heidelberg.

[112] w. Lu, J., j. He, Y., y. Li, H., and l. Lu, F. (2006). Detecting small target of ship at sea by infrared image. In *2006 IEEE International Conference on Automation Science and Engineering*, pages 165–169.

[113] Wackerman, C., Friedman, K., Pichel, W., Clemente-Coln, P., and Li, X. (2001). Automatic detection of ships in radarsat-1 sar imagery. *Canadian Journal of Remote Sensing*, **27**(5), 568–577.

[114] Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA. ACM.

[115] Wang, B. and Dudek, P. (2014). A fast self-tuning background subtraction algorithm. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[116] Wang, C., Jiang, S., Zhang, H., Wu, F., and Zhang, B. (2014a). Ship detection for high-resolution sar images based on feature analysis. *IEEE Geoscience and Remote Sensing Letters*, **11**(1), 119–123.

[117] Wang, D., Lu, H., and Yang, M. H. (2013). Least soft-threshold squares tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2371–2378.

[118] Wang, H. and Suter, D. (2005). A re-evaluation of mixture of Gaussian background modeling [video signal processing applications]. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pages ii/1017–ii/1020 Vol. 2.

[119] Wang, N. and Yeung, D.-Y. (2013). Learning a deep compact image representation for visual tracking. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 809–817. Curran Associates, Inc.

[120] Wang, N., Shi, J., Yeung, D.-Y., and Jia, J. (2015). Understanding and diagnosing visual tracking systems. In *The IEEE International Conference on Computer Vision (ICCV)*.

[121] Wang, R., Bunyak, F., Seetharaman, G., and Palaniappan, K. (2014b). Static and moving object detection using flux tensor with split gaussian models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[122] Wang, S., Wang, M., Yang, S., and Jiao, L. (2017). New hierarchical saliency filtering for fast ship detection in high-resolution sar images. *IEEE Transactions on Geoscience and Remote Sensing*, **55**(1), 351–362.

[123] Wang, Y. and Liu, H. (2012). A hierarchical ship detection scheme for high-resolution sar images. *IEEE Transactions on Geoscience and Remote Sensing*, **50**(10), 4173–4184.

[124] Weiss, J. M., Luo, R., and Welch, R. M. (1997). Automatic detection of ship

tracks in satellite imagery. In *Geoscience and Remote Sensing, 1997. IGARSS '97. Remote Sensing - A Scientific Vision for Sustainable Development., 1997 IEEE International*, volume 1, pages 160–162 vol.1.

[125] Wen, L., Cai, Z., Lei, Z., Yi, D., and Li, S. (2014). Robust online learned spatio-temporal context model for visual tracking. *Image Processing, IEEE Transactions on*, **23**(2), 785–796.

[126] White, B. and Shah, M. (2007). Automatically tuning background subtraction parameters using particle swarm optimization. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1826–1829.

[127] Wu, Y., Ling, H., Yu, J., Li, F., Mei, X., and Cheng, E. (2011). Blurred target tracking by blur-driven tracker. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1100–1107.

[128] Wu, Y., Lim, J., and Yang, M. H. (2013). Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418.

[129] Yang, X., Wang, M., and Tao, D. (2015). Robust visual tracking via multi-graph ranking. *Neurocomputing*, **159**, 35 – 43.

[130] Ycel, Z., Salah, A. A., Merili, ., Merili, T., Valenti, R., and Gevers, T. (2013). Joint attention by gaze interpolation and saliency. *IEEE Transactions on Cybernetics*, **43**(3), 829–842.

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

[132] Yu, J. G., Xia, G. S., Gao, C., and Samal, A. (2016). A computational model for object-based visual saliency: Spreading attention along gestalt cues. *IEEE Transactions on Multimedia*, **18**(2), 273–286.

[133] Zhang, J., Ma, S., and Sclaroff, S. (2014a). *MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization*, pages 188–203. Springer International Publishing, Cham.

[134] Zhang, K. and Song, H. (2013). Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*, **46**(1), 397 – 411.

[135] Zhang, K., Zhang, L., and Yang, M.-H. (2014b). Fast compressive tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **36**(10), 2002–2015.

[136] Zhang, L. and van der Maaten, L. (2014). Preserving structure in model-free tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(4), 756–769.

[137] Zhao, Y., Gong, H., Lin, L., and Jia, Y. (2008). Spatio-temporal patches for night background modeling by subspace learning. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4.

[138] Zhaoying, H. and Jinsong, C. (2004). A review of ship detection algorithms in polarimetric sar images. In *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, volume 3, pages 2155–2158 vol.3.

[139] Zhong, W., Lu, H., and Yang, M. H. (2012). Robust object tracking via sparsity-based collaborative model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845.

[140] Zivkovic, Z. and van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, **27**(7), 773 – 780.

[141] Zou, Z. and Shi, Z. (2016). Ship detection in spaceborne optical image with svd networks. *IEEE Transactions on Geoscience and Remote Sensing*, **54**(10), 5832–5845.