Extending Visual Object Tracking for Long Time Horizons

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

Master of Science
in
Electronics and Communication Engineering
by Research

by

Abhinav Moudgil
201331039
abhinav.moudgil@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2019

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Extending Visual Object Tracking for Long Time Horizons" by Abhinav Moudgil, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Vineet Gandhi

To my family.

For everything.

Acknowledgments

I would like to express my sincere gratitude to Prof. Vineet Gandhi who gave me an opportunity and freedom to work on the topics I was interested in. I am grateful for his support and I will carry forward the lessons I learnt while working with him.

I wholeheartedly thank my wing-mates and friends at IIIT-H who supported me through my thick and thin and contributed to my holistic development. A few words don't do justice to the beautiful memories and experiences I shared with you all; I will cherish them forever.

Abstract

Visual object tracking is a fundamental task in computer vision and is a key component in wide range of applications like surveillance, autonomous navigation, video analysis and editing, augmented reality etc. Given a target object with bounding box in the first frame, the goal in visual object tracking is to track the given target in the subsequent frames. Although significant progress has been made in this domain to address various challenges like occlusion, scale change etc., we observe that tracking on a large number of short sequences as done in previous benchmarks does not clearly bring out the competence or potential of a tracking algorithm. Moreover, even if a tracking algorithm works well on challenging small sequences and fails on moderately difficult long sequences, it will be of limited practical importance since many tracking applications rely on precise long-term tracking. Thus, we extend the problem of visual object tracking for long time horizons systematically in this thesis.

First, we first introduce a long-term visual object tracking benchmark. We propose a novel large-scale dataset, specifically tailored for long-term tracking. Our dataset consists of high resolution, densely annotated sequences, encompassing a duration of over 400 minutes (676K frames), making it more than 20 folds larger in average duration per sequence and more than 8 folds larger in terms of total covered duration, compared to existing generic datasets for visual tracking. The proposed dataset paves a way to suitably assess long term tracking performance and train better deep learning architectures (avoiding/reducing augmentation, which may not reflect real world behaviour). We also propose a novel metric for long-term tracking which captures the ability of a tracker to track consistently for long duration. We benchmark 17 state of the art trackers on our dataset and rank them according to several evaluation metrics and run time speeds. Next, we analyze the long-term tracking performance of state of the art trackers in depth. We focus on the three key aspects of long-term tracking: Re-detection, Recovery and Reliability. Specifically, we (a) test re-detection capability of the trackers in the wild by simulating virtual cuts, (b) investigate the role of chance in recovery of tracker post failure and (c) propose a novel metric allowing visual inference on the contiguous and consistent aspect of tracking. We present several insights derived from an extensive set of quantitative and qualitative experiments.

Lastly, we present a novel fully convolutional anchor free siamese framework for visual object tracking. Previous works utilized anchor based region proposal networks to improve the performance of siamese correlation based trackers while maintaining real-time speed. However, we show that enumerating multiple boxes at each keypoint location in the search region is inefficient and unsuitable for the task of single object tracking, where we just need to locate one target object. Thus, we take an alternate

approach by directly regressing box offsets and sizes for keypoint locations in the search region. This proposed approach, dubbed SiamReg, is fully convolutional, anchor free, lighter in weight and improves target localization. We train our framework end-to-end with Generalized IoU loss for bounding box regression and cross entropy loss for target classification. We perform several experiments on standard tracking benchmarks to demonstrate the effectiveness of our approach.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent boom in social media has established video as one of the essential means of communication and entertainment. Billions of people daily record videos and upload them online. Also, significant advancements in hardware and electronics allow us to record videos from all kinds of devices like mobile phones, drones, Kinect etc. In order to manage and index, understand and store this huge amount of video data, intelligent video systems are highly sought-after. Visual object tracking is one of the fundamental tasks in computer vision which allows us to build such intelligent video solutions. Simply put, it "aims at estimating the trajectory of an object in the image plane as it moves around a scene" [90].

Visual tracking systems must be robust enough to recognize the target object in unconstrained general settings. This still remains one of the most challenging problems in vision since it poses various practical challenges like loss of information from 3D environment to 2D video domain, occlusions, illumination variations, significant viewpoint change etc. The system is expected to work well with multi-domain data with minimal changes in the parameters of tracking framework. Despite these challenges, researchers have significantly advanced state of the art performance on standard tracking benchmarks, thanks to the boom in data and hardware which allows us to train deep learning based video recognition systems. Fast forward to today, many top performing trackers on existing benchmarks utilize deep features in their framework and even run beyond real-time speeds.

However, the deployment of these trackers in practical large-scale applications is still debatable. Most of the tracking applications like surveillance [73], traffic flow monitoring [14] etc. often require precise tracking for long duration of time. We assert that even if a tracking algorithm works well on extremely challenging small sequences from existing benchmarks but fails on moderately difficult long sequences, it will be of limited practical importance. Our analysis in Chapter 3 shows that none of the benchmarks truly evaluates the trackers from *long-term perspective*. We thus approach this problem of "long-term object tracking" systematically in this thesis. This chapter first discusses the area of visual object tracking briefly and then lists our contributions in detail along with the overall thesis outline.

1.1 Visual Object Tracking

Visual object tracking has been studied and applied in a variety of domains. Several works utilized object tracking for surveillance [37, 7, 10], sports [47], movie editing [49] or animal groups tracking [41]. Some of these applications require multiple objects or targets to be tracked. This domain in vision is typically known as multi-object tracking. Multi-object tracking is more of an "identity association" problem where we need to accurately assign identities to the detections obtained from object detection systems. Thus, the performance of multi-object tracking systems depends on the accuracy of detections obtained as well as the assignments done by the tracking algorithm. Multiple object tracking scenarios often also require domain specific information.

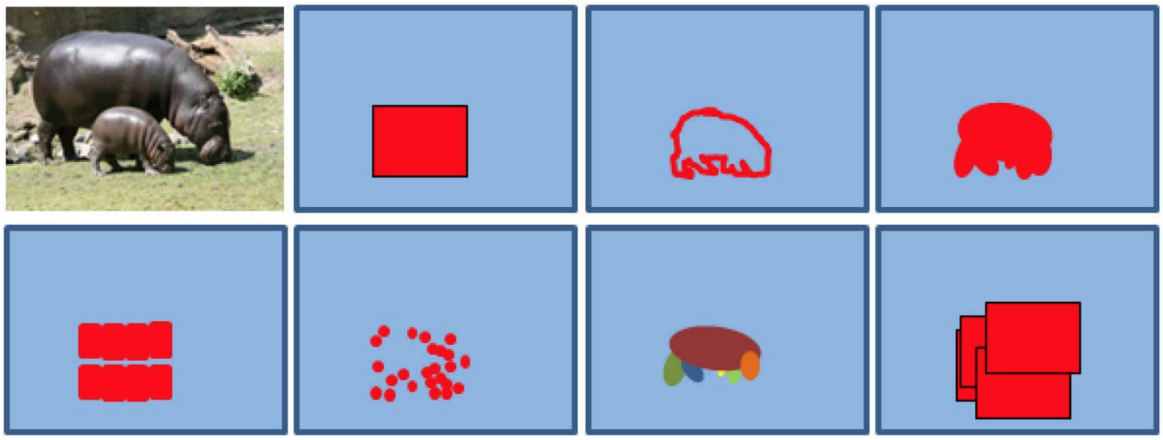

Figure 1.1 In visual tracking, target can be represented in various ways. This is an illustration from Smeulders *et al.* [74] in which 8 such ways have been presented from left to right and top to bottom: bounding box, contour, blob, patch-based, set of salient features, parts and multiple boxes.

In this thesis, we focus on the problem of single object tracking. The problem is formally defined as follows: We are given a specific target in the first frame of a video sequence and the goal is to track the given target in the subsequent video frames. The target in the first frame could be represented in multiple ways like bounding box, contour etc. as illustrated in Fig. 1.1. Bounding box is the most popular choice among these representations and we also adopt it. This choice of bounding box creates a natural question in readers mind: How is object tracking different from object detection? The key difference between object detection and object tracking is that there is no notion of "class" in object tracking. The objects in tracking do not have any specific class whereas in object detection challenges [71, 55], the objects in the image belong to some specific class. For example, in COCO object detection challenge [55], the dataset consists of 80 classes which implies that each object in the test image will belong to one of these 80 classes. Thus, the goal in object detection is to learn a generic representation of each of these object classes in order to accurately detect and assign class label to the objects.

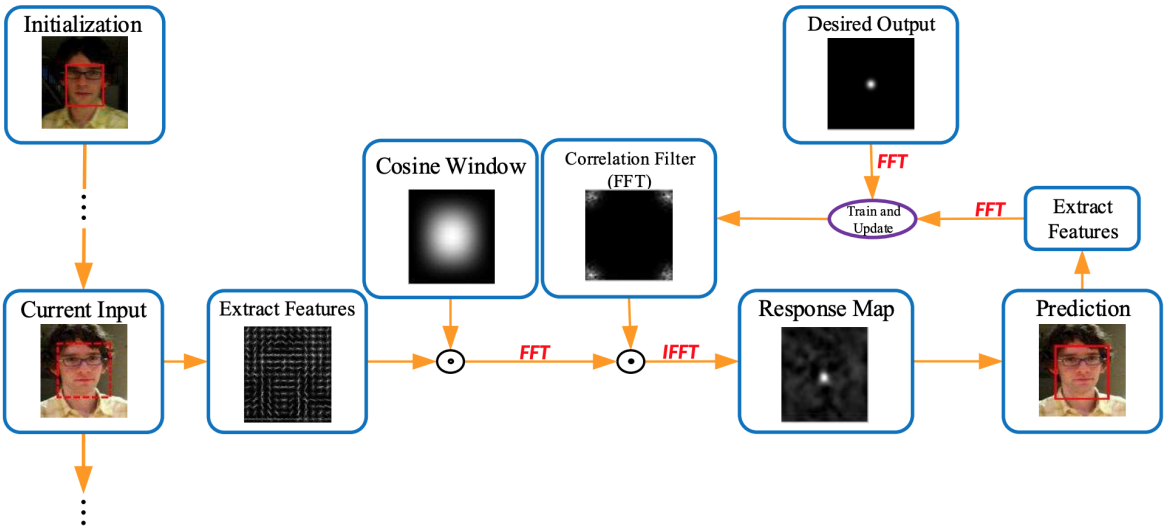

Figure 1.2 General framework of correlation filter based tracking methods [13].

On the other hand, targets in tracking do not have a specific class. This poses the challenge of data scarcity in tracking i.e. we only have a single frame to learn the representation of object to be tracked. Also, in the tracking sequence, there are several challenges which need to be handled like heavy occlusion, illumination variation, pose and scale change, motion blur, deformable objects etc. To address these challenges, researchers often use a simple cue of temporality in tracking. By exploiting this temporal cue that object representation would not change much between consecutive frames, several tracking algorithms have been proposed. The tracking algorithms have evolved a lot from simple point based KLT [80, 57] tracking in 1980s to recent deep learning based trackers [62, 51] which are much more complex.

Point based tracking: In this framework, several feature points in the image space are utilized for tracking a specific target. The target is identified by the thresholding the number of tracked feature points. Feature point tracking is involved and has been improved extensively to handle challenges like occlusions or false detections [52]. Several motion cues are utilized to track feature points such as proximity (point will not change its position much from one frame to another), mutual displacement (displacement of points in the same neighbourhood is similar) etc. Kalman [28] and Particle filter [52] are some of the popular point based object tracking frameworks. Kalal *et al.* proposed a tracking-learning-detection paradigm [40] by fusing the point tracking framework with detection to handle target disappearances and failures.

Kernel tracking: Kernel or correlation filter based tracking is another efficient paradigm in object tracking. In this setup, a correlation filter for the target object is trained from the first frame which is updated online. The target object in the subsequent frames is tracked by correlating the filter over the

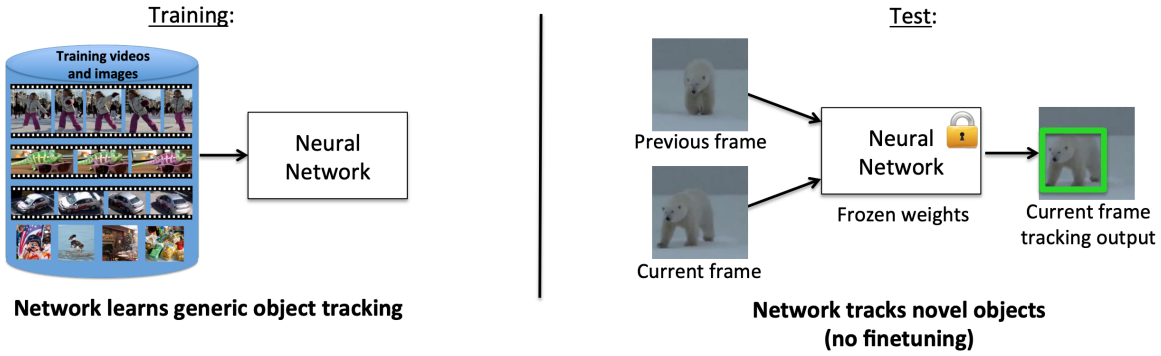

Figure 1.3 An overview of deep siamese framework for tracking proposed in [32]. A generic object recognition system is learnt offline from larget-scale data. During test time, the system is initialized with the target object in the first frame and it is tracked without any finetuning.

search area. The peak location with maximum response yields the new position of the target object. For fast tracking, all the operations like correlation, training and updating are performed in the frequency domain with fast fourier transform [8, 65]. Initial works [9, 8] used hand crafted features for correlation like HOG [16], HAAR [84] etc., however, various attempts have been made recently to utilize deep features in this setup [22, 18]. Fig. 1.2 gives a clear picture of this correlation filtering framework.

Deep convolutional tracking: The advent of deep learning significantly improved the performance on object recognition systems. AlexNet [48], the first large-scale Convolutional Neural Network (CNN) proposed by Krizhevsky *et al.* became the top performer in ImageNet object classification challenge [23] in 2012. Gradually, CNNs have been adopted in several fields of computer vision like object detection [79], semantic segmentation [64] and action recognition [3]. The main driving force behind the success of CNNs was the availability of large-scale data. Most of the object recognition systems based on CNNs require images at least in the order of 10^5. Contrastingly, in object tracking we only have a *single* frame of target object which is to be tracked in the video sequence.

To cope with the data scarcity in object tracking, Nam *et al.* first propose an effective multi-domain convolutional network [62] to learn a target-background classifier offline. Each domain is modelled by a separate branch in the last layer of the network and is representative of a particular target class. During test time or tracking, they remove all the branches in the last classifier layer of the network and replace them with a fresh branch whose weights are finetuned in online fashion as tracking proceeds further. Held *et al.* later proposed a siamese architecture which essentially learns a generic matching function [32]. As illustrated in Fig. 1.3, the system is trained offline with large-scale data of images and videos. Tuples of the same object are picked from video sequences and the target bounding box is directly regressed. Synthetic tuples are also created from static object images with augmentation. During test time, the object is tracked using the learnt siamese detector without any finetuning.

1.2 Contributions

Although several tracking algorithms discussed above work well on standard tracking benchmarks [87, 89, 61], we observe that their performance drops abruptly by several folds when they are evaluated on long sequences and their rankings of trackers also change. This highlights the fact that tracking on short sequences does not clearly demonstrate the potential of a tracking algorithm. Moreover, several applications require precise long-term tracking [49, 10]. Motivated by these observations, we present a large-scale benchmark for long-term tracking in this thesis. We discuss the causes of tracking failures in long duration such as accumulation of error or drift. We propose novel evaluation strategies which allow us to study trackers from long-term perspective both qualitatively and quantitatively. Furthermore, we present an efficient siamese framework which improves tracking precision and localization. Concretely, following are the major contributions of this thesis:

1. We introduce the long-term object tracking problem to the vision community with a benchmark. To this end, we propose a novel large-scale dataset, specifically tailored for long-term tracking. Our dataset is diverse, densely annotated, high resolution and several folds larger in length than the existing generic datasets in object tracking.

2. We benchmark 17 state of the art trackers on our proposed long-term dataset. We observe significant drop in the performance of the trackers on our dataset and we present several key insights into the challenges faced by these trackers.

3. We propose a novel metric for long-term tracking. The proposed metric captures the ability of a tracker to track consistently for long duration. We evaluate the trackers with our metric and present a thorough qualitative and quantitative analysis. Furthermore, we extend our proposed metric for visual interpretation which allows us to analyze and select the trackers as per practitioner's needs.

4. We investigate the role of chance and distractors in the recovery of trackers in long-term setting. We present a novel setup to test the re-detection capability of trackers in the wild and utilize state of the art object detection system to figure out the role of distractors in long-term tracking.

5. We propose a novel siamese framework for generic visual object tracking. The proposed tracker is fully convolutional, anchor-free, lighter in weight than the previous anchor based frameworks and runs at 204 FPS. We evaluate our tracker on standard tracking datasets to demonstrate the effectiveness of our approach.

1.3 Thesis Outline

The rest of this thesis has been organized as follows. Chapter 2 presents the background and related work of visual object tracking. Several benchmarks, evaluation strategies and tracking algorithms are

discussed. In Chapter 3, we discuss our proposed long-term tracking benchmark. We describe our long-term dataset and compare it with existing generic tracking datasets. Several baseline trackers are evaluated with the proposed long-term metric and standard tracking metrics. Chapter 4 presents an in-depth analysis of the long-term tracking performance. Specifically, it touches upon the re-detection, recovery and reliability aspect of long-term tracking. Chapter 5 describes our fully convolutional anchor free siamese framework for object tracking. A brief motivation for this framework is presented, which lead to its formulation. Several experiments are performed on the standard tracking datasets to show the effectiveness of this approach. Finally, we conclude the thesis in Chapter 6.

Chapter 2

Background and Related Work

2.1 Tracking Datasets

There are several existing datasets which are widely used for evaluating the tracking algorithms. The OTB50 [87], OTB100 [89] are the most commonly used ones. They include 50 and 100 sequences respectively and capture a generic real world scenario (where some videos are taken from platforms like YouTube and some are specifically recorded for tracking application). They provide per frame bounding box annotation and per sequence annotation of attributes like illumination variation, occlusion, deformation etc.

The ALOV300++ dataset [74] focuses on diversity and includes more than 300 short sequences (average length of only about 9 seconds). The annotations in ALOV300++ dataset are made every fifth frame. A small set of challenging sequences (partially derived from OTB50, OTB100 and ALOV300++ datasets) has been used in VOT14 [45] and VOT15 [46] datasets. They extend the rectangular annotations to rotated ones and provide per frame attribute annotations, for more accurate evaluation. Both of these datasets have been instrumental in yearly visual object tracking (VOT) challenge.

Some datasets have focused on particular type of applications/aspects. TC128 [54] was proposed to study the role of color information in tracking. It consists of 128 sequences (some of them are common to OTB100 dataset) and provides per frame annotations and sequence wise attributes. Similarly, UAV [61] targets the tracking application, when the videos are captured from low-altitude unmanned aerial vehicles. The focus of their work is to highlight challenges incurred while tracking in video taken from an aerial viewpoint. They provide both real and synthetically generated UAV videos with per frame annotations.

More recently, two datasets were proposed to incorporate the benefits of advances in capture technology. The NFS [26] dataset was proposed to study the fine grained variations in tracking by capturing high frame rate videos (240 FPS). Their analysis shows that since high frame video reduces appearance variation per frame, it is possible to achieve state of the art performance using substantially simpler tracking algorithms. Another recent dataset called AMP [95], explores the utility of 360° videos to generate and study tracking with typical motion patterns (which can be achieved by varying the camera

re-parametrization in omni-directional videos). Contemporary to our work, [58] and [83] also review recent trackers for long-term tracking. However, they limit the long-term tracking definition to the ability of a tracker to re-detect after object goes out of view and the quality of their long term datasets is lower than our proposed TLP dataset, in terms of resolution and per sequence length. We evaluate the trackers from a holistic perspective and show that even if there is no apparent major challenge or target disappearance, tracking consistently for a long period of time is an extremely challenging task.

Although recent advances pave the way to explore several novel and specific fine grained aspects, the crucial long term tracking aspect is still missing from most of the current datasets. The typical average length per sequence is still only about 10-30 seconds. The proposed TLP dataset takes it to about 8-9 minutes per sequence, making it the largest densely annotated high-resolution dataset for the application of visual object tracking.

2.2 Tracking Methods

Long duration tracking still remains challenging, however, here we list some notable attempts which led to significant progress. Collins *et al.* [15] proposed the idea of using neighbourhood around the ground truth for discriminative feature learning. This idea was later extended into tracking by detection frameworks [1]. Kalal *et al.* [39] proposed TLD framework of learning detector from initial tracking, maintaining confidence of local tracking based on feature point tracks and switching to detection in low confidence scenarios. TLD tracker was one of the first attempts to neatly handle the re-detection problem, which is crucial for long term tracking. The consistency aspect of tracking was then improved by employing an ensemble of classifiers [93] instead of a single one. These methods maintain several weak classifiers, often initiated at different checkpoints to account for appearance variations of the target.

Another currently popular direction is discrimative correlation filter based tracking *et al.* [8, 21]. These methods exploit the properties of circular correlation (efficiently performed in Fourier domain) for training a regressor in a sliding-window fashion. Major gains were achieved by integrating multi resolution shallow and deep features maps to learn the correlation filters [18, 6, 81]. Another fundamental contribution is the use of siamese networks for visual object tracking [5, 32]. The GOTURN tracker [32] uses the siamese architecture to directly regress the bounding box locations given two cropped images from previous and current frames. On the other hand the SiamFC tracker [5] transforms the exemplar image and the large search image using the same function and outputs a map by computing similarity in the transformed domain. It is illustrated in Fig. 2.1. These efforts [5, 32] can be seen as learning an offline similarity function and since they do not involve any online updates, they are extremely efficient in terms of computation.

Another noteworthy effort came from Nam *et al.* [63] (Fig. 2.2), which introduced the idea of treating the tracking problem as classifying candidate windows sampled around the previous target region. Other interesting ideas include using adversarial training for tracking [77] or casting object

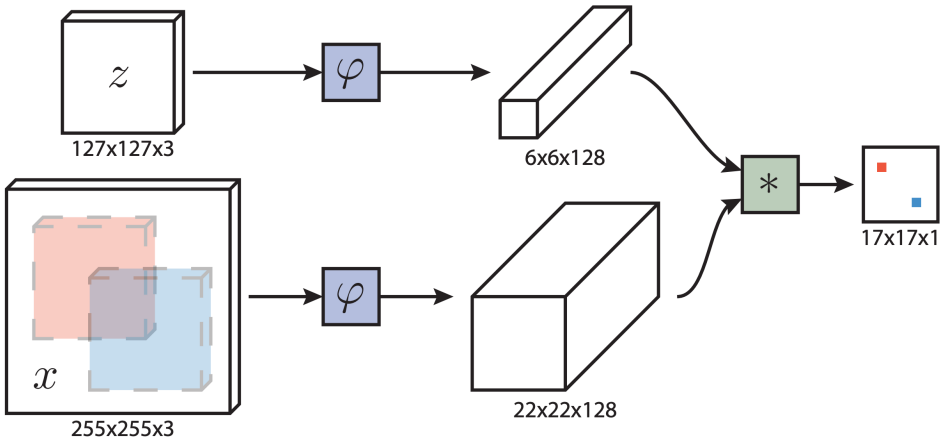

Figure 2.1 Overview of fully convolutional siamese framework [5] for visual object tracking.

tracking as a Q-learning problem [91, 12]. Some recent efforts borrow ideas from the object detection literature and demonstrate their benefits for the task of tracking. The Real-time MDNet[38] tracker uses RoIAlign which was introduced in Mask R-CNN [30]. The work by Li *et al.* [50, 51] augments the features extracted through a siamese networks with Region Proposals Networks (RPN) for classification and regression. The RPN branch brings significant improvements over the previous siamese trackers [5] in terms of accurate prediction of scale and aspect ratio of the bounding boxes.

2.3 Tracking Metrics

Early works relied on the precision metric[1, 88] for quantifying the tracking performance, which computes the pixel distance between the center of the ground truth and the prediction. This was convenient since it required only annotating the center of the target and not the whole bounding box. However, since this does not account for the scale and aspect ratio, the success metric[88] was introduced. It measures the percentage of frames where the Intersection Over Union (IOU) of the predicted and ground truth bounding boxes is more than a threshold. Failure rate [43] was then introduced to address the tracking length measure. In this measure, a manual operator reinitializes the tracker upon every failure. The number of required manual interventions per frame is recorded as the quantitative measure. It is indicative of the continuity and consistency aspect of tracking, however, due to required manual interventions, it is unscalable for long sequences (in large datasets). For a more detailed review and analysis of metrics for short-term tracking, we would refer the reader to the work by Cehovin *et al.* [11].

A few evaluations metrics have been proposed targeting the long duration tracking. Valmadre *et al.* [83] introduced True Positive Rate (TPR), True Negative Rate (TNR) and took their geometric mean. To have a single representative metric accounting for the trackers which do not predict absent labels,

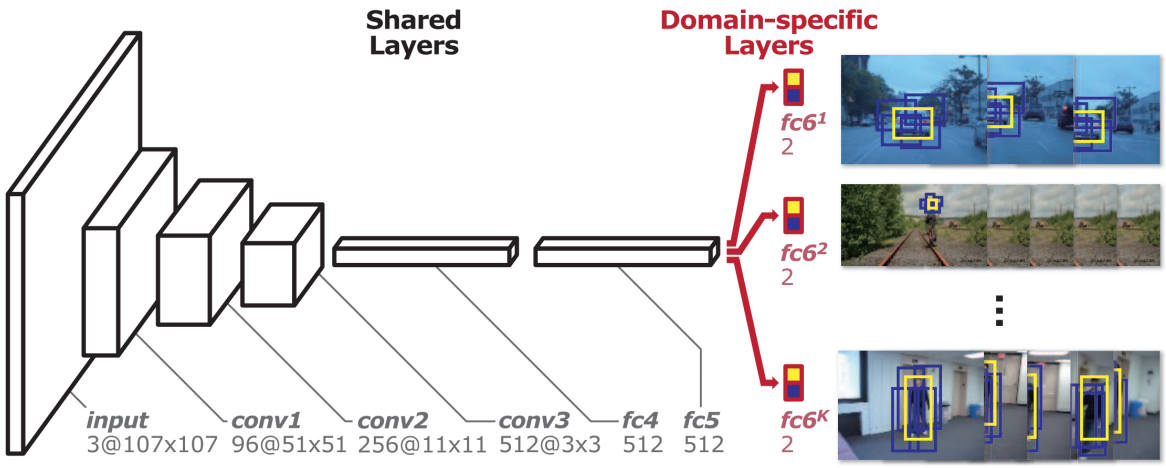

Figure 2.2 Multi-domain Convolutional Network for object tracking [62]. The network consists of domain specific branches in the last layer and the first 5 convolutional layers are shared across all the branches. This helps to separate target specific information from generic object representation.

they proposed a modified metric called maximum geometric mean metric. However, the metric is biased towards the ability of a tracker to predict absent labels.

Lukezic *et al.* [58] introduced tracking recall and precision and use this to give a tracking F1 score. However, their definition of a long term tracker is limited to the ability of a tracker to predict absence and the proposed metric does not focus on the continuity and consistency aspect of tracking. We believe the ability to track for long duration consistently even when the target object is always present has been overlooked in these previous efforts [58, 83]. Lukezic *et al.* also proposed an experiment to quantify the re-detection ability of a tracker. However, their experiment essentially focuses on the search strategy with no appearance changes. Here, we seek to quantify the re-detection ability in the wild. Moudgil and Gandhi [60] proposed the Longest Subsequence Measure (LSM), which quantifies the longest contiguous segment successfully tracked in the sequence. Here, we propose an extension of it called 3D-LSM, which allows to compare trackers visually.

Chapter 3

Long-Term Visual Object Tracking Benchmark

This chapter aims to emphasize the fact that tracking on large number of tiny sequences, does not clearly bring out the competence or potential of a tracking algorithm. Moreover, even if a tracking algorithm works well on extremely challenging small sequences and fails on moderately difficult long sequences, it will be of limited practical importance. Many tracking applications like surveillance, autonomous navigation, video analysis and editing, augmented reality etc. of these applications rely on long-term tracking, however, only few tracking algorithms have focused on the challenges specific to long duration aspect [40, 59, 34, 78]. Although they conceptually attack the long term aspect, the evaluation is limited to shorter sequences or couple of selected longer videos. The recent correlation filter [19, 22, 4, 94] and deep learning [92, 62, 5, 32] based approaches have significantly advanced the field, however, their long term applicability is also unapparent as the evaluation is limited to datasets with typical average video duration of about 20-40 seconds. Not just the evaluation aspect, the lack of long term tracking datasets has been a hindrance for training in several recent state of the art approaches. These methods either limit themselves to available small sequence data [62, 92] or use augmentation on datasets designed for other tasks like object detection [32].

Motivated by the above observation, we propose a new long duration dataset called Track Long and Prosper (TLP), consisting of 50 long sequences. The dataset covers a wide variety of target subjects and is arguably one of the most challenging datasets in terms of occlusions, fast motion, viewpoint change, scale variations etc. However, compared to existing generic datasets, the most prominent aspect of TLP dataset is that it is larger by more than 20 folds in terms of average duration per sequence, which makes it ideal to study challenges specific to long duration aspect. For example, drift is a common problem in several tracking algorithms and it is not always abrupt and may occur due to accumulation of error over time (which may be a slow procedure and can be difficult to gauge in short sequences). Similarly, long sequences allow us to study the consistency of a tracker to recover from momentary failures.

We select 17 recent state of the art trackers which are scalable to be evaluated on TLP dataset and provide a thorough evaluation in terms of tracking accuracy and real time performance. Testing on such a large dataset significantly reduces the overfitting problem, if any, and reflects if the tracker is actually designed to consistently recover from challenging scenarios. To present a further perspective,

	Frame rate (FPS)	# videos	Min Duration (sec)	Mean Duration (sec)	Max Duration (sec)	Total Duration (sec)
UAV123 [61]	30	123	3.6	30.5	102.8	3752
OTB50 [87]	30	51	2.3	19.3	129	983
OTB100 [89]	30	100	2.3	19.6	129	1968
TC128 [54]	30	129	2.3	14.3	129	1844
VOT14 [45]	30	25	5.7	13.8	40.5	346
VOT15 [46]	30	60	1.6	12.2	50.2	729
ALOV300 [74]	30	314	0.6	9.2	35	2978
NFS [26]	240	100	0.7	16	86.1	1595
TLP	24/30	50	**144**	**484.8**	**953**	**24240**

Table 3.1 Our proposed TLP dataset consists of 50 HD densely annotated sequences from real world scenarios, encompassing a duration of over 400 minutes (676K frames). The sequences in our dataset are much longer in duration (8 minutes or 484 seconds) than the sequences in other tracking datasets.

we provide a comprehensive attribute wise comparison of different tracking algorithms by selecting various sets of short sequences (derived from original TLP sequences), in which each set only contains sequences where a particular type of challenge is dominant (like illumination variation, occlusions, out of view etc.).

We observe that the rankings from previous short sequence datasets like OTB50 [87] significantly vary from the rankings obtained on the proposed TLP dataset. Several top ranked trackers on recent benchmarks fail to adapt to long-term tracking scenario and their performance drops significantly. Additionally, the performance margin notably widens among several trackers, whose performances are imperceptibly close in existing benchmarks. More specifically, apart from MDNet [62], performance of all other evaluated tracker drops below 25% on commonly used metric of area under the curve of success plots. Our investigation hence strongly highlights the need for more research efforts in long term tracking and to our knowledge the proposed dataset and benchmark is the first systematic exploration in this direction.

3.1 TLP Dataset

The TLP dataset consists of 50 videos collected from YouTube. The dataset was carefully curated with 25 indoor and 25 outdoor sequences covering a large variety of scene types like sky, sea/water, road/ground, ice, theatre stage, sports arena, cage etc. Tracking targets include both rigid and deformable/articulated objects like vehicle (motorcycle, car, bicycle), person, face, animal (fish, lion, puppies, birds, elephants, polar bear), aircraft (helicoptor, jet), boat and other generic objects (e.g sports ball). The application aspect was also kept into account while selecting the sequences, for example we include long sequences from theatre performances, music videos and movies, which are rich in content, and tracking in them may be useful in context of several recent applications like virtual camera simulation or video stabilization [49, 29]. Similarly, long term tracking in sports videos can be quite helpful for automated analytics [56]. The large variation in scene type and tracking targets can be observed in

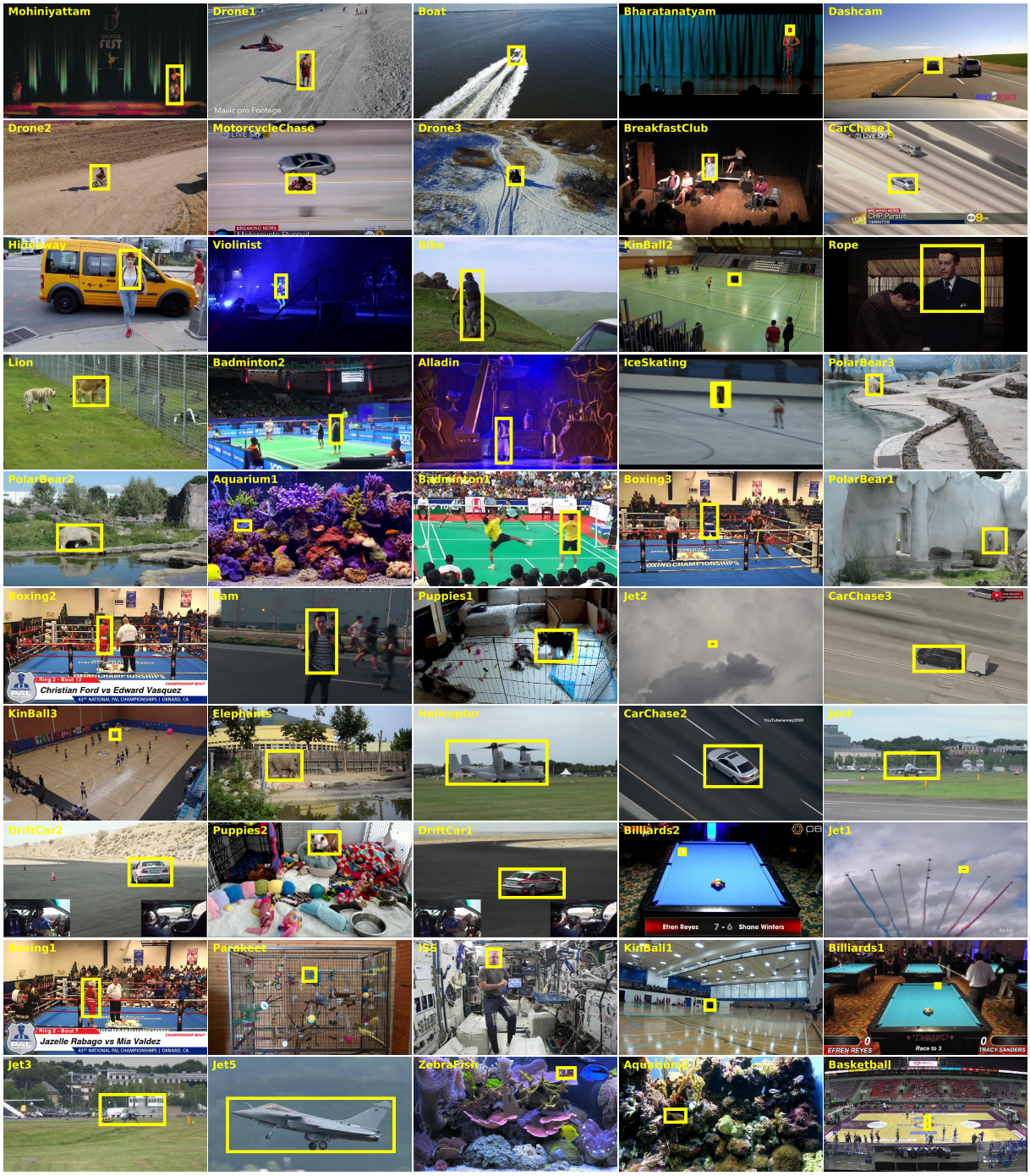

Figure 3.1 First frames of all the 50 sequences of TLP dataset. The sequences are sorted in ascending order on the basis of mean success rate (defined in Section 3.2) of all trackers at IoU threshold of 0.5. The sequences at the bottom right are more difficult to track than the ones at the top left.

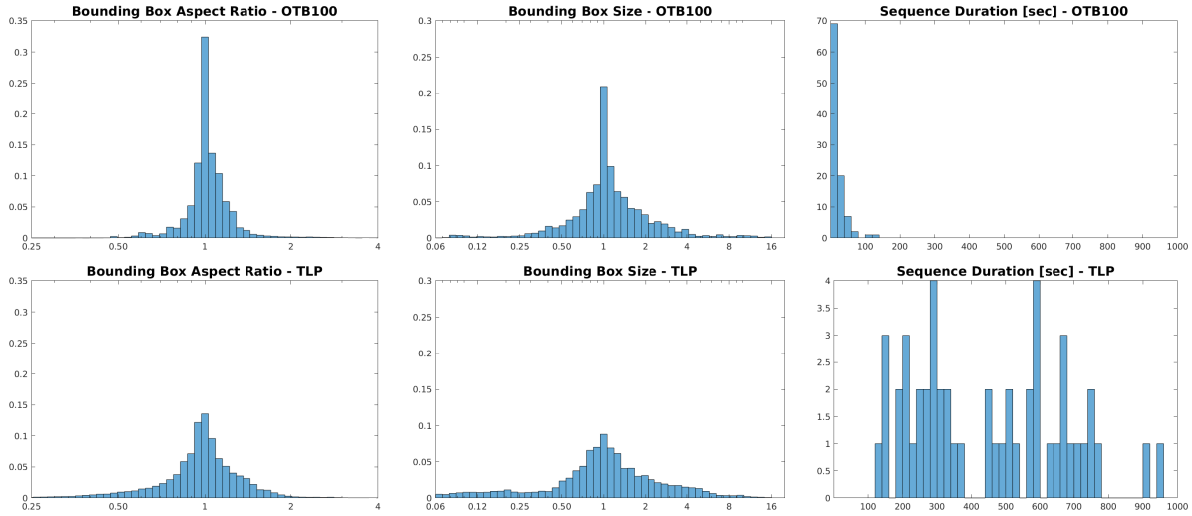

Figure 3.2 Column 1 and 2: Proportional change of the targets aspect ratio and bounding box size (area in pixels) with respect to the first frame in OTB100 and TLP. Results are compiled over all sequences in each dataset as a histogram with log scale on the x-axis. Column 3: Histogram of sequence duration (in seconds) across the two datasets.

Figure 3.1. We further compare the TLP dataset with OTB in Figure 3.2, to highlight that the variation in bounding box size and aspect ratio with respect to the initial frame is significantly larger in TLP and the variations are also well balanced. The significant differences in duration of sequences in OTB and TLP are also apparent.

The per sequence average length in TLP dataset is over 8 minutes. Each sequence is annotated with rectangular bounding boxes per frame, which were done using the VATIC [85] toolbox. The annotation format is similar to OTB50 and OTB100 benchmarks to allow for easy integration with existing toolboxes. We have 33/50 sequences (amounting to 4% frames in total) in TLP dataset where the target goes completely out of view and thus, we provide absent label for each frame in addition to the bounding box annotation. All the selected sequences are single shot (do not contain any cut) and have a resolution of 1280×720. Similar to VOT [46], we choose the sequences without any cuts, to be empirically fair in evaluation, as most trackers do not explicitly model a re-detection policy. However, the recovery aspect of trackers still gets thoroughly evaluated on the TLP dataset, due to presence of full occlusions and out of view scenarios in several sequences.

TinyTLP and TLPattr: We further derive two short sequence datasets from TLP dataset. The TinyTLP dataset consists of first 600 frames (20 sec) in each sequence of the TLP dataset to compare and highlight the challenges incurred due to long-term tracking aspect. The length of 20 sec was chosen to align the average per sequence length with OTB100 benchmark. The TLPattr dataset consists of total 90 short sequences focusing on different attributes. Six different attributes were considered in our work i.e (a) fast motion of target object or camera, (b) illumination variation around target object between consecutive frames, (c) large scale variation of the target object, (d) partial occlusions of the

target object by other objects or background, (e) out of view or full occlusions, where object leaves the camera view or it is not visible at all and (f) background clutter. The TLPattr dataset includes 15 short sequences corresponding to each of the attribute.

Each sequence in TLPattr is carefully selected in such a way that the only dominant challenge present in it is a particular attribute, it is assigned to. For example, for fast motion, we first select all instances in entire TLP dataset where the motion of the center of the ground truth bounding box between consecutive frames is more than 20 pixels. We temporally locate every such fast motion event and curate a short sequence around it by selecting 100 frames before and after the fast motion event. We then sort the short sequences based on the amount of motion (with the instance with most movement between two frames as the top sequence) and manually shortlist 15 sequences (starting from the top), where fast motion is the only dominant challenge present and simultaneously avoiding selection of multiple short sequences from the same long video. For attributes like illumination variation and background clutter the selection was fully manual. The rationale behind curating the TLPattr dataset was the following: (a) Giving a single attribute to entire sequence (as in previous works like OTB50) is ill posed on long sequences as in TLP. Any attribute based analysis with such an annotation would not capture the correct correlation between the challenge and the performance of the tracking algorithm. (b) Using per frame annotation of attributes is also difficult for analysis in long videos, as the tracker may often fail before reaching the particular frame where attribute is present and (c) The long sequences and variety present in TLP dataset allows us to single out a particular attribute and choose subsequences where that is the only dominant challenge. This paves the way for accurate attribute wise analysis.

3.2 Evaluation

3.2.1 Evaluated Algorithms

We evaluated 17 recent trackers on the TLP and TinyTLP datasets. The trackers were selected based on three broad guidelines i.e.: (a) they are computationally efficient for large scale experiments; (b) their source codes are publicly available and (c) they are among the top performing trackers in existing benchmarks. Our list includes CF trackers with hand crafted features, namely SRDCF [21], MOSSE [8], DCF [33], DSST [20], KCF [33], SAMF [53], Staple [4], BACF [42] and LCT [59]; CF trackers with with deep features: ECO [19] and CREST [75] and deep trackers i.e. GOTURN [32], MDNet [62], ADNet [92] and SiamFC[5]. We also included TLD [40] and MEEM [93] as two older trackers based on PN learning and SVM ensemble, as they specifically target the drift problem for long-term applications. We use default parameters on the publicly available version of the code when evaluating all the tracking algorithms.

	MDNet	SiamFC	CREST	ADNet	GOTURN	ECO	MEEM	BACF	TLD	SRDCF	STAPLE	SAMF	DSST	LCT	DCF	KCF	MOSSE
TinyTLP	83.4	70.1	65.8	68.7	51.8	57.6	49.2	60.0	36.4	54.1	60.0	58.9	56.5	42.7	41.6	41.3	37.2
TLP	42.1	27.6	24.9	22.1	22.0	21.8	19.5	15.9	13.8	13.2	13.1	11.3	8.8	8.7	7.9	6.9	3.7

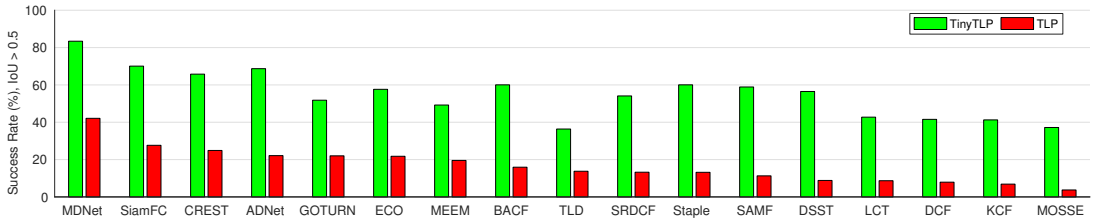

Figure 3.3 Success rate of individual trackers on TinyTLP and TLP datasets. The algorithms are sorted from left to right based on their performance on TLP.

3.2.2 Evaluation Methodology

We use precision plot, success plot and longest subsequence measure for evaluating the algorithms. The precision plot [2, 87] shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. A representative score per tracker is computed, by fixing a threshold over the distance (we use the threshold as 20 pixels). The success metric [87] computes the intersection over union (IoU) of predicted and ground truth bounding boxes and counts the number of successful frames whose IoU is larger than a given threshold. In out of view scenarios, if the tracking algorithm explicitly predicts the absence, we give it an overlap of 1 otherwise 0. The success plot shows the ratio of successful frames as the IoU threshold is varied from 0 to 1. A representative score for ranking the trackers is computed as the area under curve (AUC) of its success plot. We also employ the conventional success rate measure, counting frames above the threshold of 0.50 (IoU $>$ 0.50).

LSM metric: We further propose a new metric called Longest Subsequence Measure (LSM) to quantify the long term tracking behaviour. The LSM metric computes the ratio of the length of the longest successfully tracked continuous subsequence to the total length of the sequence. A subsequence is marked as successfully tracked, if $x\%$ of frames within it have IoU > 0.5, where x is a parameter. LSM plot shows the variation in the normalized length of longest tracked subsequence per sequence, as x is varied. A representative score per tracker can be computed by fixing the parameter x (we use the threshold as 0.95).

The LSM metric captures the ability of a tracker to track continuously in a sequence within a certain bound on failure tolerance (parameter x) and bridges the gap over existing metrics which fail to address the issue of frequent momentary failures. For example, it often happens in long sequences that tracker loses the target at some location and freezes there. If coincidentally the target passes the same location (after a while), the tracker starts tracking it again. LSM penalizes such scenarios by considering only the longest continuous tracked subsequences.

3.2.3 Per Tracker Evaluation

Table 3.2.3 presents the success rate of each individual tracker on TinyTLP and TLP datasets. The MDNet tracker is the best performing tracker on both the datasets. TLD is the worst performing tracker on TinyTLP and MOSSE performs worst on TLP dataset. The performance significantly drops for each tracker on TLP dataset, when compared to TinyTLP dataset, which clearly brings out the challenges incurred in long-term tracking. The relative performance drop is minimum in MDNet where the success rate reduces from 83.4% to 42.1% (roughly by a factor of 2) and is most in MOSSE tracker, which reduces from 37.2% in TinyTLP to 3.7% in TLP (reduction by more than a factor of 10).

In general, the relative performance decrease is more in CF trackers with hand crafted features as compared to CF+deep trackers. For instance, trackers like BACF, SAMF, Staple give competitive or even better performance than CREST and ECO over TinyTLP dataset, however, their performance steeply decreases on TLP dataset. Although all the CF based trackers (hand crafted or CNN based) are quite susceptible to challenges such as long term occlusions or fast appearance changes, our experiments suggest that using learnt deep features reduces accumulation of error over time and reduces drift. Such accumulation of error is difficult to quantify in short sequences and the performance comparison may not reflect the true ability of the tracker. For example, BACF outperforms ECO on TinyTLP by about 2%, however it is 6% worse than ECO on TLP. Similarly, the performance difference of SAMF and ECO is imperceptible on TinyTLP, which differs by almost a factor of 2 on TLP.

The deep trackers outperform other trackers on TLP dataset, with MDNet and SiamFC being the top performing ones. ADNet is third best tracker on TinyTLP, however, its performance significantly degrades on TLP dataset. It is interesting to observe that both MDNet and ADNet refine last fully connected layer during online tracking phase, however, MDNet appears to be more consistent and considerably outperforms ADNet on TLP. The offline trained and freezed SiamFC and GOTURN perform relatively well (both appearing in top five trackers on TLP), however SiamFC outperforms GOTURN, possibly because it is trained on larger amount of video data. Another important observation is that the performance of MEEM surpasses all state of the art CF trackers with hand crafted features on TLP dataset. The ability to recover from failures also allows TLD tracker (giving lowest accuracy on TinyTLP) to outperform several recent CF trackers on TLP.

3.2.4 Overall Performance

The overall comparison of all trackers on TinyTLP and TLP using Success plot, Precision plot and LSM plot are demonstrated in Figure 3.2.4. In success plots, MDNet clearly outperforms all the other trackers on both TinyTLP and TLP datasets with AUC measure of 68.1% and 36.9% respectively. It is also interesting to observe that the performance gap significantly widens up on TLP and MDNet clearly stands out from all other algorithms. This suggests that the idea of separating domain specific information during training and online fine tuning of background and foreground specific information, turns out to be an extremely important one for long term tracking. Furthermore, analyzing MDNet and

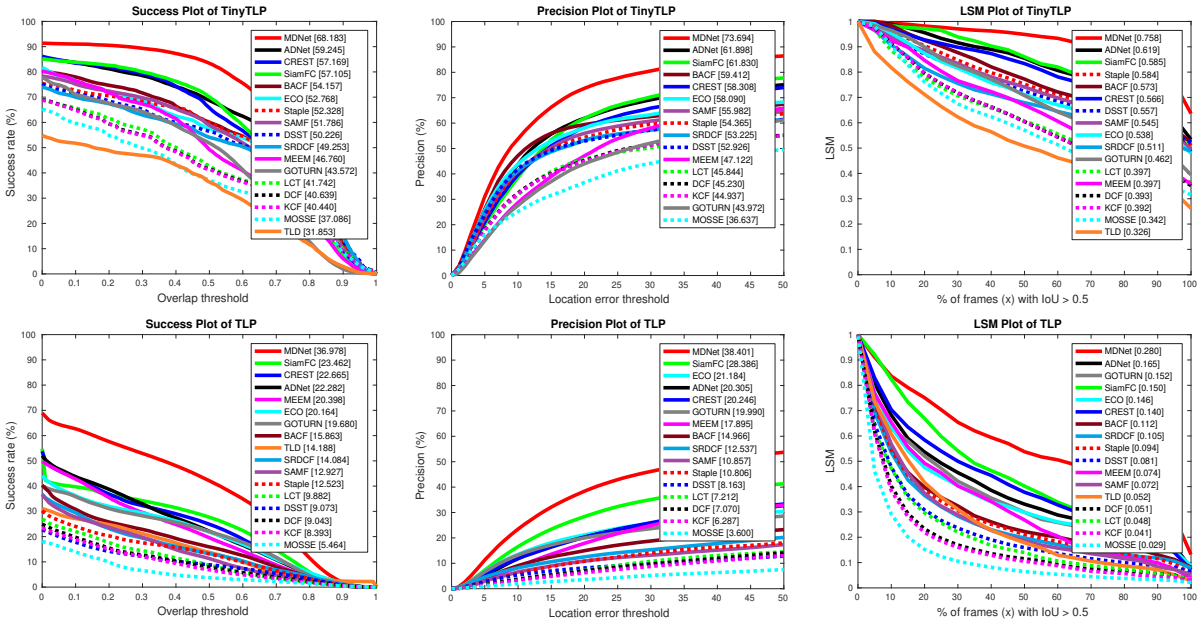

Figure 3.4 Overall Performance of evaluated trackers on TinyTLP and TLP with success plot, precision plot and LSM plot respectively (each column). For each plot, ranked trackers are shown with corresponding representative measure i.e. AUC in success plots; 20 pixel threshold in precision plots and 0.95 as length ratio in LSM plots.

ADNet both of which employ the strategy of online updates on last FC layers during tracking, it appears that learning to detect instead of learning to track gives a more robust performance in long sequences. The performance drop of SiamFC and GOTURN on TLP also suggests a similar hypothesis.

The steeper success plots in TLP as compared to TinyTLP dataset, suggest that accurate tracking gets more and more difficult in longer sequences, possibly due to accumulation of error. The lower beginning point on TLP (around 40-50% for most trackers compared to 80-90% on TinyTLP), indicates that most trackers entirely drift away before reaching halfway through the sequence. The rankings in success plot on TLP are also quite contrasting to previous benchmarks. For instance, ECO is the best performing tracker on OTB100 closely followed by MDNet (with almost imperceptible difference), and its performance significantly slides on TLP. Interestingly, MEEM breaks into top five trackers in AUC measure of success plot on TLP (ahead of ECO). In general there is striking drop of performance between TinyTLP and TLP for most CF based trackers (more so for hand crafted ones). CREST is most consistent among them and ranks in top 5 trackers for both TinyTLP and TLP.

The precision plots also demonstrate similar trends as success plots, however they bring couple of additional subtle and interesting perspectives. The first observation is that SiamFC's performance moves closer to performance of MDNet on TLP dataset. Since SiamFC is fully trained offline and does not make any online updates, it is not accurate in scaling the bounding box to the target in long term, which brings down its performance in IoU measure. However, it still hangs on to the target due to the large

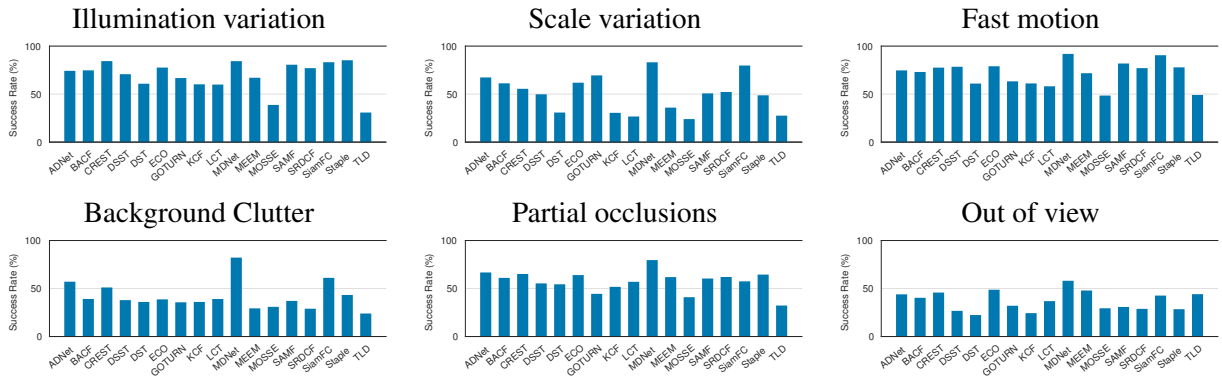

Figure 3.5 Attribute wise performance evaluation on TLPattr dataset. Results are reported as success rate (%) with IoU > 0.5.

scale training to handle challenges in predicting the consecutive bounding boxes, hence the numbers improve in the precision plot (again precision plot on TinyTLP does not capture this observation). The ADNet tracker is ranked two on TinyTLP using precision measure, however, it drops to 4th position on TLP. The GOTURN tracker also brings minor relative improvement in precision measure and moves ahead of MEEM on TLP.

The LSM plots show the ratio of longest successfully tracked continuous subsequence to the total length of the sequence. The ratios are finally averaged over all the sequences for each tracker. A sequence is successfully tracked if $x\%$ of frames in it have IoU > 0.5. We vary the value x to draw the plots and the representative number is computed by keeping $x = 95\%$. This measure explicitly quantifies the ability to continuously track without failure. MDNet performs the best on this measure as well. The relative performance of CREST drops in LSM measure, as it partially drifts away quite often, however is able to recover from it as well. So its overall success rate is higher, however, the average length of longest continuous set of frames it can track in a video is relatively low. In general, the ratio of largest continuously tracked subsequence to sequence length (with success rate > 0.95) averaged over all sequences is about 1/4th for MDNet and lower than 1/6th for other trackers. This indicates the challenge in continuous accurate tracking without failures.

3.2.5 Attribute wise Performance Evaluation

The average attribute wise success rates of all the trackers on TLPattr dataset are shown in Figure 3.5. Each attribute in TLPattr dataset includes 15 short sequences corresponding to it (dominantly representing the particular challenge). Out of view appears to be the most difficult challenge hindering the performance of the trackers followed by background clutter, scale variation and partial occlusions. Most of the trackers seem to perform relatively better on sequences with illumination variation and fast motion. On individual tracker wise comparison, MDNet gives best performance across all the attributes, clearly indicating the tracker's reliable performance across different challenges.

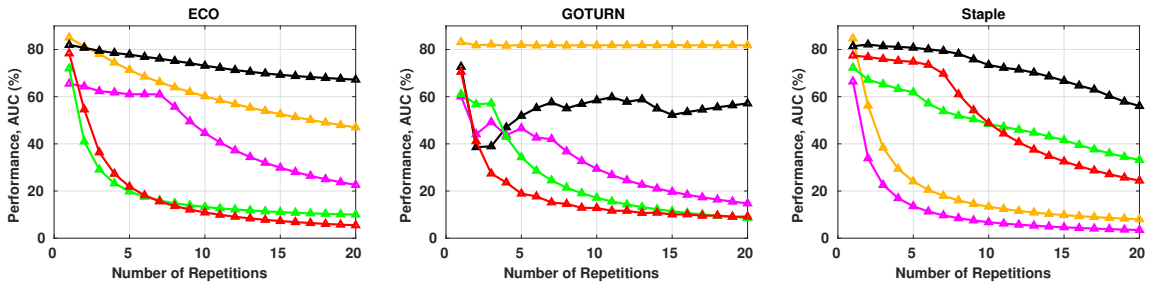

Figure 3.6 Results of three different trackers on 20 times elongated TinyTLP sequences (by reversing and concatenating the sequence in iterative way). Each color represents a different sequence and each triangle represents a repetition.

Another important perspective to draw from this experiment is that the analysis on short sequences (even if extremely challenging) is still not a clear indicator of their performance on long videos. For example, Staple and CREST are competitive in performance across all the attributes, however their performance on full TLP dataset differs by almost a factor of two in success rate measure (CREST giving a value 24.9 and Staple is only 13.1). Similarly comparison can be drawn between DSST and GOTURN, which are competitive in per attribute evaluation (with DSST performing better than GOTURN on fast motion, partial occlusions, background clutter and illumination variation). However, in long terms setting, their performance varies by a large margin (GOTURN giving success rate of 22.0, while DSST is much inferior with a value of 8.8).

3.3 Ablation Studies

3.3.1 Evaluation on repeated TinyTLP sequences

The essence of our paper is the need to think "long term" in object tracking, which is crucial for most practical applications of tracking. However, it remains unclear if there exists a "long term challenge in itself" and one can always argue that the performance drop in long videos is just because of "more challenges" or "frequent challenges". To investigate this further, we conduct a small experiment where we take a short sequence and repeat it 20 times to make a longer video out of it, by iteratively reversing and attaching it at the end to maintain the continuity. This increases the length of the sequence without introducing any new difficulty or challenges. In Figure 3.6, we present such an experiment with three different trackers ECO (deep+CF, best performing tracker on OTB), GOTURN (pure deep) and Staple (pure CF) on 5 TinyTLP sequences for each tracker, where the tracker performs extremely well in the first iteration. We can observe that the tracking performance degrades for all three algorithms (either gradually or steeply) as the sequences get longer, which occurs possibly due to error accumulated over time. This again highlights the fact the tracking performance not just depends on the challenges present in the sequence but also gets affected by the length of the video. Hence, a dataset like TLP, inter-

leaving the challenges and the long term aspect, is necessary for comprehensive evaluation of tracking algorithms.

3.3.2 Runtime analysis

The run time speeds of all the evaluated algorithms are presented in Figure 3.7. For fair evaluation, we tested all the CPU algorithms on a 2.4GHz Intel Xeon CPU with 32GB RAM and we use a NVIDIA GeForce GTX 1080 Ti GPU for testing GPU algorithms. The CF based trackers clearly are most computationally efficient and even CPU algorithms run several folds faster than real time. The deep CF and deep trackers are computationally more expensive. MDNet gives lowest tracking speeds and runs at 1 FPS even on GPU. Among deep trackers GOTURN is the fastest tracker, however SiamFC and ADNet bring a good trade off in terms of overall success rate and run time speeds on GPU.

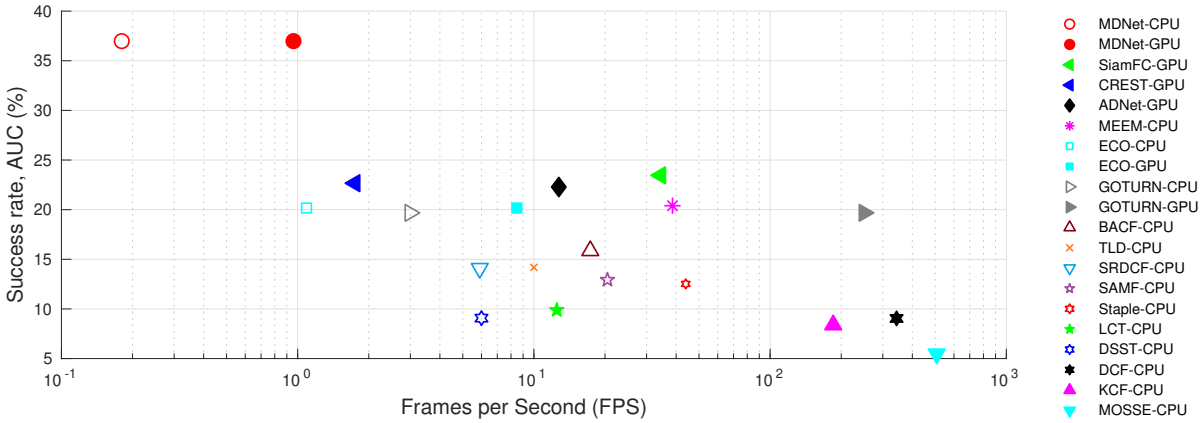

Figure 3.7 Runtime comparison of different tracking algorithms.

3.4 Summary

We propose the TLP dataset, focusing on the long term tracking application, with notably larger average duration per sequence, a factor which is of extreme importance and has been neglected in the existing benchmarks. We evaluate 17 state of the art algorithms on the TLP dataset, and the results clearly demonstrate that almost all state of the art tracking algorithms do not generalize well on long sequence tracking, MDNet being the only algorithm achieving more than 25% on the AUC measure of success plots. However, MDNet is also the slowest among the evaluated 17 trackers in terms of run time speeds.

Interestingly, if we only select the first 20 seconds of each sequence for evaluation (calling it TinyTLP dataset), the performance of all the trackers increases by multiple folds across different metrics. Another important observation is that the evaluations on small datasets fail to efficiently discriminate the performances of different tracking algorithms, and closely competing algorithms on TinyTLP result in

quite different performance on TLP. The dominant performance of MDNet suggests that the ideas of online updating the domain specific knowledge and learning a classifier cum detector instead of a tracker (which regresses the shift), are possibly some cues to improve the performance in long term setting. Our evaluation on repeated TinyTLP sequences shows that temporal depth indeed plays an important role in the performance of evaluated trackers and appropriately brings out their strengths and weaknesses. To the best of our knowledge, TLP benchmark is the first large-scale evaluation of the state of the art trackers, focusing on long duration aspect and makes a strong case for much needed research efforts in this direction, in order to track long and prosper.

Chapter 4

Analyzing Long-term Tracking Performance

Deep learning advances like in other fields have significantly improved the state of the art in object tracking. However, the question which remains is that how close we are to practitioners needs, requiring consistent and reliable long duration tracking. Interestingly, most existing works evaluate their performance on datasets consisting of multiple short clips. For instance, the most commonly used OTB dataset has average length of about 20 seconds [88] per clip. Recent work [60] highlights that same trackers (working well on short sequences) when evaluated on long sequences lead to significant performance drop. Following works [25, 83, 58] also make similar observations and suggest that we need alternate ways to evaluate and analyze long term tracking performance.

These works [60, 25, 83, 58] indicate that three properties are crucial for an improved long term tracking performance. First, is the ability to re-detect the target if it is lost. This is crucial to handle situations where the target object goes out of the frame and reappears. It is also important to re-initiate tracking when the target object is lost due to occlusions or momentary tracking failures. The second key aspect is the ability of the tracker to distinguish between the actual target and distractor or background clutter. This aspect is important in consistent tracking as well as recovering from failures. Figure 4.1 illustrates an example where chance plays a major role in recovery. It is necessary to scrutinize the nature of failures and recoveries, to better understand and improve long term tracking performance. The third key aspect is the reliability which connects to the ability for consistent and continuous tracking. This suggests the ability of the tracker to track for long duration without failures. Tracking in long duration video allows us to study factors like slow accumulation of error which may be difficult to observe in short sequences. Several applications like video surveillance or virtual camera simulation from static camera [27] require precise tracking for long time. Surprisingly, none of the current evaluation strategies account for these three crucial aspects of re-detection, recovery and reliability.

For instance, the most prevalent metrics are Success and Precision plots, which measure the number of frames with Intersection Over Union (IoU) greater than threshold and the mean distance from the center of the ground truth respectively. Both these metrics does not reflect anything specific about re-detection, consistency or the distractor discrimination. Recent work by Lukezic *et al.* [58] studied the efficacy of the search region expansion strategy of different trackers. However, the re-detection

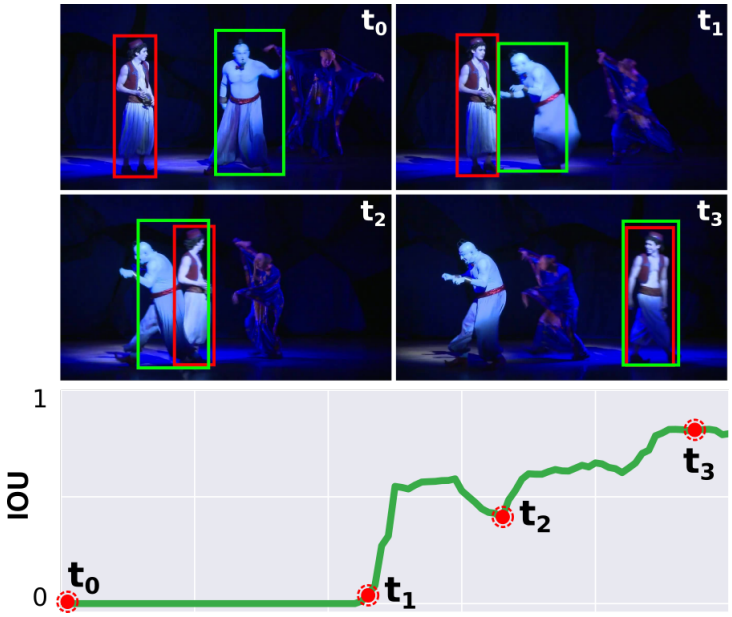

Figure 4.1 A typical example of a chance based recovery in Alladin sequence from TLP [60] dataset. SiamRPN (green) is tracking the incorrect object and has zero overlap with the target (red) in the start. It switches to tracking the target when they pass through each other. We study such chance based recoveries in long-term setting both qualitatively and quantitatively. Best viewed in colour.

experiment in their work is synthetically designed by padding with gray values which cannot be extrapolated to real world scenarios. Valmadre *et al.* [83] improve the evaluation strategy by explicitly handling the cases where the target is not visible/absent from the frame. Other recent efforts [60, 25] identify the aforementioned key issues but they do not provide any way to evaluate these properties comprehensively.

In this chapter, we propose two novel evaluation metrics focused on the re-detection ability and the aspect of continuous and consistent long term tracking. Furthermore, we present deeper insights into the failure and recovery of different trackers, explicitly addressing the role of distractors. Since, shorter sequences are inappropriate to address these concerns, we use the Track Long and Prosper (TLP) [60] dataset for the experiments. The major advantage of TLP is that the average sequence length is longest among other densely annotated datasets [35, 25, 58]. Long duration videos present cases of multiple failures and recoveries for each video, which allows for a better analysis of the re-detection, recovery and reliability aspects of tracking.

As our first contribution, we propose a re-detection experiment by simulating cuts (an abrupt transition from a frame to another) in long videos. Cuts are introduced by minimizing the Generalized IoU [70] between the ground truth bounding boxes in the frame before and after the cut. Different trackers are then evaluated on their ability to recover/re-detect and the time they take to recover. As our second contribution we formally study the chance factor in recoveries post failure. More specifically,

we analyze the role of distractors in failures and recoveries and the co-incidences which aid tracking. For example, it often happens in long sequences that tracker loses the target at some location and freezes there. If by chance the target passes the same location (after a while), the tracker starts tracking it again. Our study aims to quantify such behaviour. As our third contribution, we propose 3D Longest Subsequence Measure (3D-LSM), as a metric for quantifying the consistency of tracking. The 3D-LSM metric quantifies the performance of a tracker by measuring the longest contiguous sequence successfully tracked at a given precision and allowed failure tolerance. The 3D-LSM also allows for a direct visual interpretation of tracking results in the form of a 2D image.

4.1 Re-detection in the Wild

This experiment is specifically designed to evaluate a tracker's ability to re-detect the object after it is lost. Ability to re-detect is crucial in long-term tracking because the target may go out of the view or a tracker could lose the target due to momentary failures. We seek to quantify the re-detection ability of a tracker in a real-world scenario.

4.1.1 Setup

We select a segment from a sequence, and delete it, thereby introducing a cut. This is illustrated in Figure 4.2. We evaluate the tracker's performance on the segment after the cut to judge the re-detection ability of the tracker. We choose the segment for a cut in each sequence from TLP dataset, such that the Generalized IoU [70] between the target bounding boxes before and after the cut is minimized. By GIoU, we are able to capture the "distance" between bounding boxes in a generic way which implicitly takes into account various factors like center distance, scale and aspect ratio. The duration of the cut is fixed to 300 frames. We empirically find that it is a good balance between having the target move far away from the tracker's search region without significantly varying the other aspects in the scene. Keeping the similar context around the target helps to keep the focus on the re-detection ability (the context can change a lot in long sequences if the length of the omitted sequence is large). This scheme is very general and can be applied on datasets which have no target disappearances at all.

4.1.2 Evaluation

For a fair re-detection experiment, the tracker is initialized with the target annotation 100 frames before the cut. We choose 100 frames so that the tracker starts stable tracking before the cut. It also allows trackers with online updates to build a reasonable representation of the target object. We also make sure that there are no critical challenges in this duration of 100 frames such as heavy occlusion, clutter etc. After the cut, the tracker is continued to run on the sequence for another 200 frames and its performance on this segment is evaluated. We define "recovery" when the IoU of the tracker with the

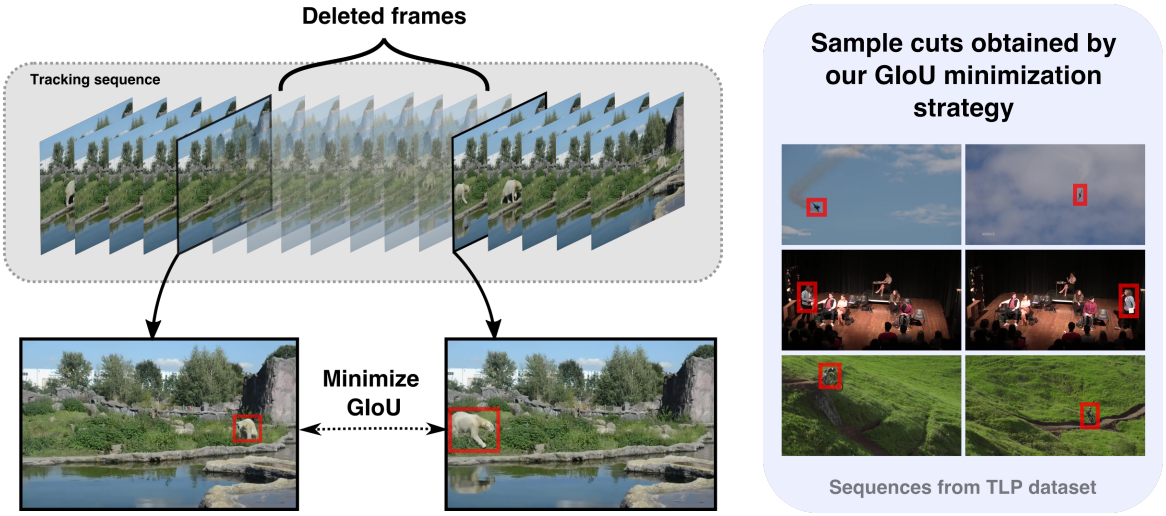

Figure 4.2 A cut is introduced by removing a set of contiguous frames from a tracking sequence. This introduces a sudden change of position of the ground truth object as shown in the left diagram. The red bounding box shows the position of the target object, before and after the cut. We maximize the amount of target shift by minimizing the GIoU [70] between these bounding boxes. We evaluate the trackers ability to re-detect the object after the cut. Few more examples from TLP dataset with simulated cuts are shown on the right.

target reaches 0.5. To make a relative comparison of the trackers on the re-detection task, we report the following metrics.

- Total number of sequences (out of the total 50 TLP sequences) in which a tracker is able to recover within the remaining 200 frames.

- Total number of sequences where the recovery is "quick" i.e the recovery happens within 30 frames (1 second).

- Average number of frames a tracker takes to successfully recover.

We perform this experiment on TLP dataset with the following trackers: ATOM [17], MDNet [63], SiamRPN [51], ECO [18], LCT [59] and TLD [39]. LCT and TLD are long-term trackers with explicit re-detection ability; ATOM is the current top performing tracker on the long-term benchmark LaSOT, while MDNet, SiamRPN and ECO are the top performing trackers on other benchmarks [60, 86, 44]. This selection presents all the prevalent tracking approaches: correlation filter based trackers [18, 59], end to end classification with online updates [63], offline trained siamese trackers with region proposals [51], low level feature tracking with online learned detector [39] and combining offline and online components [17]. The same set of trackers are used in all the following experiments as well.

Tracker	Quick recov. ↑	Total recov. ↑	Recov. length (# frames) ↓
ATOM [17]	**12 / 50**	**25 / 50**	34
TLD [39]	6 / 50	10 / 50	**8**
MDNet [63]	5 / 50	13 / 50	48
ECO [18]	4 / 50	7 / 50	28
SiamRPN [51]	2 / 50	7 / 50	39
LCT [59]	2 / 50	7 / 50	143

Table 4.1 Number of quick recoveries, total recoveries and average recovery length is reported for each tracker. All the trackers are evaluated on 50 sequences of TLP dataset, augmented by our GIoU minimization cut strategy.

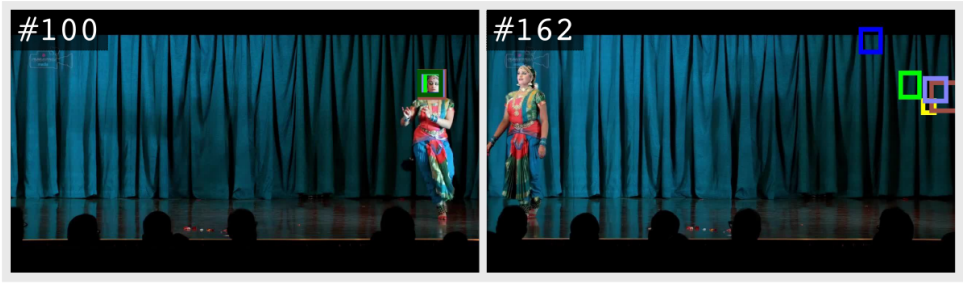

Figure 4.3 The figure illustrates a simulated cut in the Bharatanatyam sequence from TLP dataset. The cut can be seen as a representation of a situation where the performer exits the stage and enters from another end. None of the evaluated trackers was able to recover in this sequence, even with the exact same background and a single target object.

4.1.3 Results

Our results are summarized in Table 4.1. The key observation is that all trackers are unable to recover in more than half of the sequences. Quick recoveries are even fewer. Interestingly, trackers fail to recover even in absence of any distractors and minimal background change. Once such example is illustrated in Figure 4.3. ATOM performs best in our re-detection experiment, by re-detecting target on half of the sequences and takes 34 frames on average for recovery. TLD is the fastest to recover (average 8 frames), however, it only succeeds to re-detect in 10 sequences. Since TLD re-detects targets by processing low-level image features, it fails to recover targets with appearance changes like pose, view point etc. LCT also has an explicit detection module however its performance is contrasting to TLD and it takes the longest time to recover. Other trackers like ECO, MDNet and SiamRPN are limited by their search range and only recover if the target object comes within their search range after the cut. ATOM on the other hand uses a larger search area (25 times the area of target object bounding box) and hence recovers more often.

4.2 Recovery by Chance

In this section, we investigate the role of chance in tracker recovery pose failure. Interestingly, most of the evaluation metrics does not take this into account and we believe that to design better long-term trackers, it is important to scrutinize the nature of recovery. More specifically, we analyze two scenarios that frequently occurs in long sequences (a) the tracker starts tracking an alternate object and recovers back when it interacts with the target and (b) tracker freezes somewhere in the background and resumes tracking when the target passes through it.

4.2.1 Recovery by Tracking Alternate Object

The first class of recoveries we investigate is when the recovery occurs while the tracker has been stuck tracking an alternate object (distractor). We consider distractor of both the same class as well as alternate classes. The recovery here occurs only because of the interactions between the objects in the scene. An example of this kind of recovery is illustrated in Fig 4.1.

However, directly evaluating the role of distractors is challenging because single object tracking benchmarks [86, 60, 44] do not have annotations for multiple objects. We exploit the effectiveness of modern object detectors to resolve this concern. While an object detector would not be accurate enough to treat it as a ground truth for bounding boxes for alternate objects, it would still allow us to draw useful insights. Moreover, the results may vary when a different object detector is being used. Hence, the evaluation presented in this section is not intended to serve as a metric. Nonetheless, it presents important insights into the role of distractors in tracking performance, which are further highlighted by qualitative results presented in the supplementary material.

We select 16 out of the 50 sequences from TLP dataset where distractors are present and the target interacts with them. We run YOLOv3 [68, 67] on these sequences to obtain all object annotations. We compute and study the following aspects:

- **Distractor Tracking Length (DTL):** Mean percentage of frames in a sequence where the tracker is tracking (IoU ≥ 0.5) an alternate object and has zero overlap with the target (averaged over the selected 16 sequences).

- **Average Distractor Recovery (ADR):** The recoveries that occur while the tracker is tracking an alternate object (IoU with alternate object ≥ 0.5). We define recovery if the IoU with the ground truth becomes nonzero and maintains a non zero value for next 60 frames. We present average number of recoveries per sequence due to distractor tracking for each tracker.

- **Success without any Distractor Recovery (Success-DR):** The performance drop that occurs if we zero out the performance after the first instance of such a recovery. We report the original success metric (% of frames with IoU ≥ 0.5)(Success) and success metric without any distractor recovery (Success-DR) on TLP dataset.

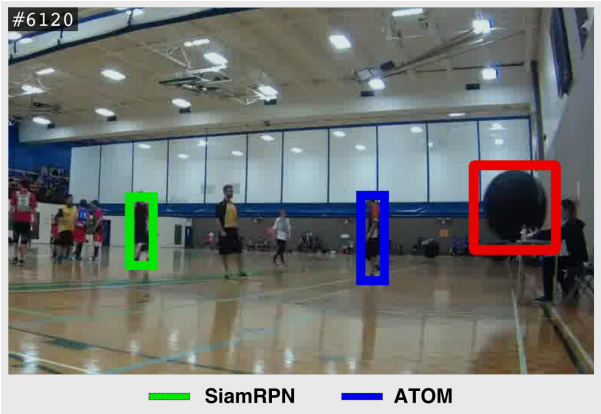

Figure 4.4 An example from TLP [60] Kinball1 sequence where the tracking target (red) is black ball. Both SiamRPN (green) and ATOM (blue) end up tracking objects of totally different class i.e. human which is also significantly different in appearance from the given target.

Our results are shown in Table 4.2. We can see that that SiamRPN and ATOM accurately track an incorrect object for more than 13% on average in a sequence, which is an exceedingly high number. This probably stems from their design which looks for "objectness" i.e. the potential bounding boxes in the neighbourhood. Surprisingly, the tracker gets confused among classes that are radically different in appearance. For instance, as shown in Figure 4.4 it starts tracking a human instead of a ball. The behaviour is mitigated in ECO, MDNet and LCT because online updates of the appearance model, which helps them learn better discriminative behaviour towards background objects. TLD gets least affected by the distractors, again due to online updates and its explicit ability to predict absent labels.

Tracker	DTL	ADR	Success	Success-DR
ATOM [17]	13.73%	5.68	31.42	18.81
SiamRPN [51]	14.92%	5.18	43.80	25.19
MDNet [63]	1.57%	0.68	40.77	38.45
ECO [18]	3.8%	1.5	22.38	19.68
LCT [59]	1.38%	0.81	11.21	10.42
TLD [39]	0.58%	0.12	7.02	4.32

Table 4.2 We report Distractor Tracking Length (DTL), Average number of Distractor Recoveries (ADR), Success metric and Success metric without any distractor recovery (Success-DR) on TLP dataset. All these metrics have been defined precisely in Section 4.2.1.

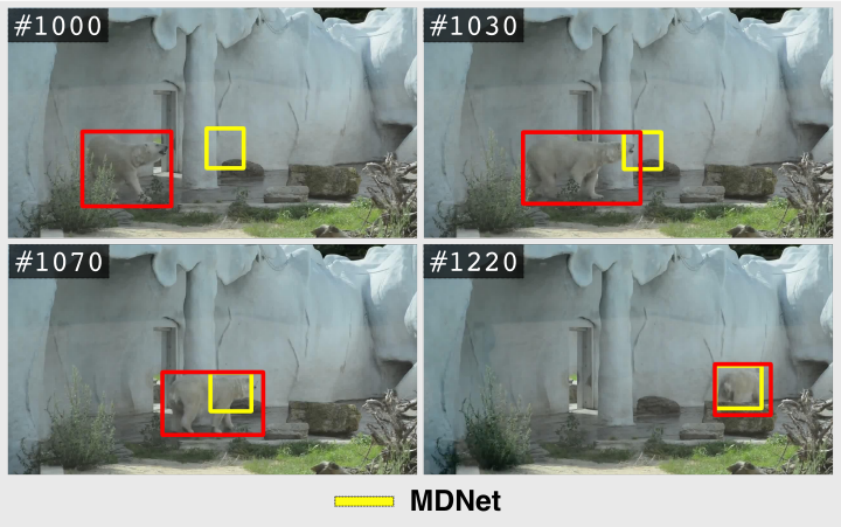

Figure 4.5 An example of a static recovery where the tracker is stationary at a particular location. Tracking only resumes when the target object passes through it.

In the last two columns of Table 4.2, we present the success metric of the listed trackers on the selected 16 sequences and the reduced performance computed by setting the IoU scores to zero post the first chance based recovery. The reduced performance is indicative of the worst case performance i.e. if a chance based recovery never happened. We observe significant drop in case of ATOM and SiamRPN. The performance drop for other trackers is also significant in context of their overall tracking performance (for example, TLD's performance drops by more than 35%). Another perspective is that ATOM and SiamRPN recover more often (second column, Table 4.2) from the chances they get and also takes good advantage of them (third and fourth column, Table 4.2). However, stronger appearance models would be crucial to improve their performance in long-term setting.

4.2.2 Recovery with No Motion

The second type of recoveries we study is when the tracker is stationary and the target passes through it and then the tracking resumes. An example of such a recovery is illustrated in Figure 4.5. Here, the recovery can be attributed to chance, because the target fortunately moved into the tracker (the tracker recovers even though it had no idea where the target was).

We first formalize the notion of the tracker being "stationary". A tracker said to be stationary, if the IoU of the current prediction (at time t) is more that 0.5 with each of the previous 200 predictions

and the IoU with the target is zero. This definition ensures that the tracker is frozen somewhere in the background, after accounting for minor noisy movements. We further define "static recovery" i.e the recovery which happens when the tracker is stationary (IoU between the tracker and target goes from zero to non-zero and remains non-zero for next 60 frames). We then compute the following aspects:

- **Average Static Recoveries (ASR):** The average number of static recoveries per sequence in the dataset.

- **Average Static Chance (ASC):** The average number of chances i.e. number of times when the tracker was stationary and the target came towards it leading to a non zero IoU (even for a single frame). In some of these chance cases, tracker actually starts tracking the target, leading to a static recovery. Thus, ASR is a subset of ASC.

- **Static Recovery Sequences (SRS):** Number of sequences (out of 50 sequences from TLP dataset) in which the tracker was stationary and it recovered through static recovery.

- **Success without any Static Recovery (Success-SR):** The impact of static recoveries on the tracking performance i.e. the reduced success metric by ignoring the performance after the first static recovery in each sequence (Success-SR). However, here we report the performance drops only on the sequences where static recovery occurs i.e. SRS and it differs for each tracker. The point of reporting these performance drops is not to give a metric, but to understand the worst case impact of such recoveries on the tracking performance.

The results are summarized in Table 4.3. The first two columns present the average number of static recoveries per sequence against the number of chances it got (averaged over all 50 sequences). The third column presents the number of sequences for each tracker which have static recoveries (the experiments are performed on all 50 sequences of the dataset, however, not all sequences have static recoveries). The last two columns present the success metric before and after accounting for the chance based recoveries (averaged only over the sequences with static recoveries, which is different for each tracker).

We see that trackers like MDNet and ECO which had a lower tendency for tracking an alternate object, have a much higher tendency to get into stationary state. ECO is most susceptible to clutter and often freezes in the background. It fails to recover successfully even after getting large number of chances. Also it is interesting to note that while SiamRPN and ATOM have the fewest chances for static recoveries, they have the steepest drop in performance when the performance after the recovery is ignored. This suggests that they are able to make better use of the recoveries than MDNet, ECO and LCT. The stationary behaviour in TLD often occurs due to drift in feature tracks, which confuses it with the background. Furthermore, we observe some contrasting nature between the two kinds of chance based recoveries discussed in our work i.e. SiamRPN and ATOM have a higher chance to track an alternate object but show less stationary behaviour, while the MDNet, ECO and LCT are opposite in nature.

Tracker	ASR	ASC	SRS	Success	Success-SR
ATOM [17]	0.98	8.06	12 / 50	25.08	16.12
SiamRPN [51]	0.58	2.22	9 / 50	40.00	21.22
MDNet [63]	3.16	15.64	13 / 50	15.14	10.64
ECO [18]	4	24.92	19 / 50	8.52	5.13
LCT [59]	3.34	7.18	21 / 50	9.66	7.15
TLD [39]	2.56	5.32	16 / 50	7.35	2.37

Table 4.3 Static recovery study: We report Average number of static recoveries per sequence (ASR), Average number of static chances a tracker gets per sequence (ASC), Sequences with Static Recoveries (SRS), Success metric and Success metric without any static recovery (Success-SR) on TLP dataset. All these metrics have been defined precisely in Section 4.2.2.

4.3 Reliability in Long-term Tracking

Practically, trackers are reliable to use in long-term applications if the human effort to fix the incorrect tracker predictions is minimal. The human effort is a function of the precision required for the application at hand. A tracker which gives contiguous segments of precise tracking would be easier to correct by re-initializing on failures. However, it will take a lot of mental burden to correct a tracker whose IoU fluctuates intermittently. We made an effort to quantify the reliability aspect and proposed the Longest Subsequence Measure (LSM) metric in the previous Chapter 3. In this section, we address some of limitations of LSM metric and extend it in a more general sense. We also present a visual interpretation of trackers which could aid the practitioner to pick appropriate trackers conditioned on their specific needs.

4.3.1 Preliminaries

LSM [60] computes the ratio of the length of the longest successfully tracked continuous subsequence to the total length of the sequence. A subsequence is marked as successfully tracked, if $x\%$ of frames within it have IoU > 0.5, where x is a slack parameter. A representative LSM score per tracker is computed by fixing the slack parameter x to 0.95.

We believe that the choice of thresholds for IoU (0.5) and slack x (0.95) in LSM does not provide a fair and complete perspective. For example, a tracker which has IoU slighly lesser than 0.5 would be penalized harshly due to binary IoU thresholding at 0.5. Prior work [72] has also shown that human annotators cannot often distinguish between IoU scores of 0.3 and 0.5. The authors also present LSM plots by fixing IoU to 0.5 and varying the slack. However, such plots fail to give a holistic perspective on the simultaneous effect of changing both the IoU and the slack.

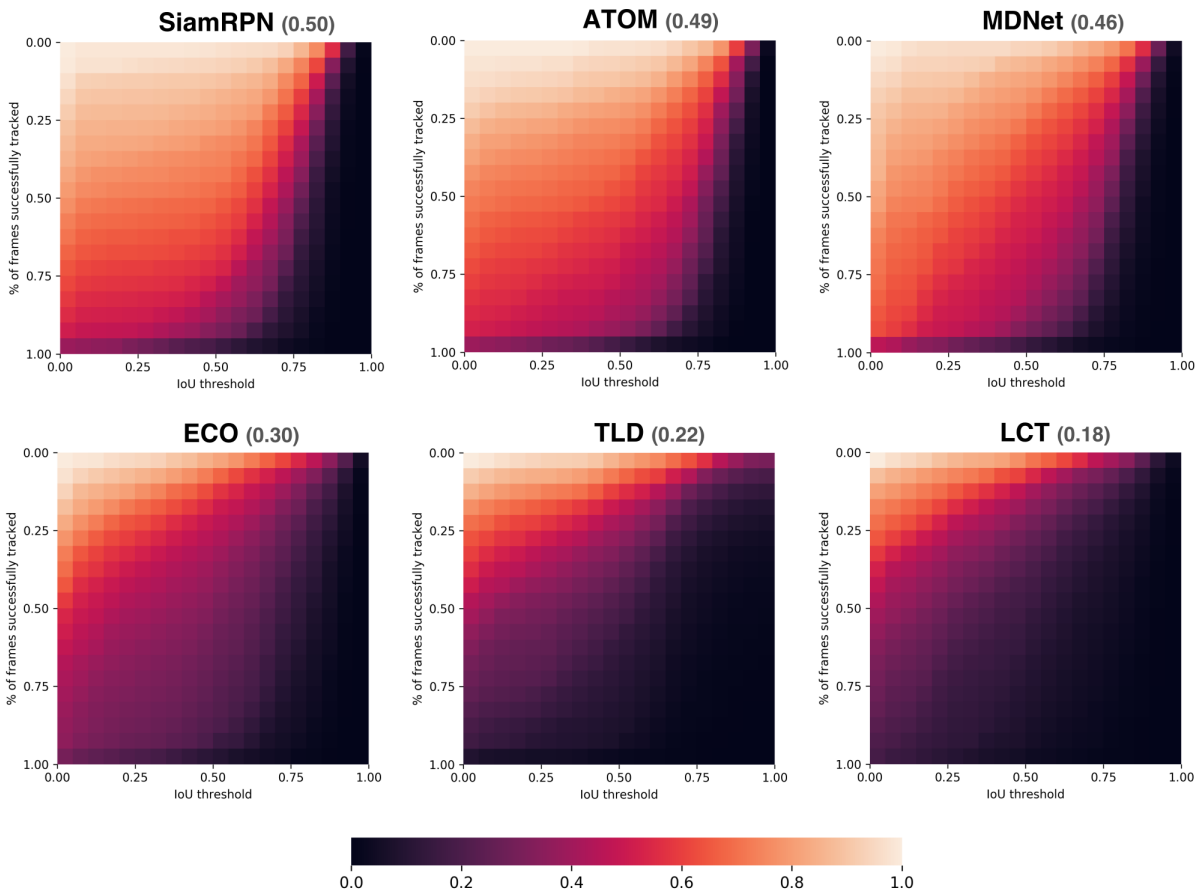

Figure 4.6 3D-LSM visualizations for the evaluated trackers. 3D-LSM metric is also reported for each tracker (on top).

4.3.2 Extending LSM

We present a 3D-LSM metric, which captures the effect of both precision (IoU) and failure tolerance in a connected manner. The 3D-LSM metrics is the mean of a matrix, computed by varying both the slack and the IoU parameters. Each entry in the matrix measures the longest contiguous subsequence (normalized) successfully tracked by fixing the IoU and slack parameters (for instance if the slack is 0.95 and IoU is 0.3, then we find the longest sub-sequence where 95% of the frames are tracked with IoU greater than 0.3). Basically, each entry in the matrix is the LSM value computed at a specific slack and IoU threshold. In current experiment we vary both slack and IoU thresholds at a rate of 0.05 from 0.05 to 1, resulting in a 20×20 matrix. One major benefit of the proposed metric is that it can be visualized as an image and makes way for a direct visual interpretation. It would aid non-expert practitioners to compare several trackers by visual inference.

4.3.3 Discussion

Our 3D-LSM visualization results for the evaluated trackers on TLP dataset are shown in Figure 4.6. SiamRPN, ATOM and MDNet significantly outperform the other three counterparts. One interesting observation is that while ECO outperforms SiamRPN on short term benchmarks like OTB100, it performs significantly worse in the presented long term setting. The figure also allows several direct visual inferences: (a) brighter plots indicate a better performance. We can observe how the images get darker when moving from SiamRPN to LCT. (b) Contours formed in better performing trackers tend to stretch towards the bottom right corner. Compare SiamRPN and ECO for instance, we can see that the shape of the contour inverts. (c) The practitioners need lies in the bottom right corner (i.e. low failure tolerance and high IoU) and most trackers are pitch black in that area. This highlights the significant challenges and opportunities which lies ahead in the area of visual object tracking to meet the application requirements.

4.4 Summary

In this paper we propose a fresh perspective on the analysis of long term tracking. We touch upon the Re-detection, Recovery and the Reliability aspects of visual object tracking, which are crucial in long-term setting. Our experiments show that most trackers are weak in reliability aspect (as shown in 3D-LSM experiments) and stronger appearance models (even the top trackers confuse between significantly varying classes) and better search strategies are a couple of cues which can help the cause. Our analysis is aimed to provide a deeper understanding of the performance of the state of the art trackers on these important aspects which are not explicit in existing evaluation metrics. The area of visual object tracking has significant advanced over the past few years, especially after the onset of deep learning algorithms. We are seeing a surge of long term benchmarks to address the widely varying application scenarios and we believe such finer analysis would pave the way for designing better long term tracking algorithms.

Chapter 5

Fully Convolutional Anchor Free Siamese Framework

Advances in deep learning has significantly improved the accuracy of tracking algorithms. Deep features deployed in correlation filtering based trackers [22, 19] achieved state of the art results on popular tracking benchmarks like OTB50 and OTB100. Nam *et al.* proposed a deep CNN based tracker MDNet [62], where they effectively learned a target-background classifier for generic objects through offline training. During testing, the last layer of CNN is fine-tuned for the specific tracking target through backpropogation and it beats the previous state of the art results by a wide margin. Despite the remarkable results, the aforementioned trackers which finetuned the features through backpropagation had one major drawback - tracking speed. Most of them operated at around $\sim$ 1 FPS, which hinders the practitioners to deploy them in applications requiring real-time tracking.

Siamese network based trackers [32, 5, 51] recently gained a lot of attention due to their high accuracy and real-time speeds. Held *et al.* first proposed a deep siamese framework GOTURN [32] to learn a generic matching between the target in previous frame and the target in current frame. The features from previous and current image frames are merged with a few fully connected layers to directly yield the target bounding box. The system is simply trained offline on large objection detection dataset and it doesn't require any online fine-tuning, thus yielding real-time speeds (100 FPS). The tracking speed comes at some cost here (a) the prediction becomes noisy when there is some occlusion involved (b) once the tracker has lost the target, it could not recover back since the target in the previous frame would not be present and it would be tracking some background in the current frame (c) the network has to *learn* translation invariance during offline training since the fully connected layers in the architecture does not preserve the spatial properties of the image.

Luca *et al.* addressed the above concerns by proposing a fully convolutional siamese framework [5]. Features of template and search images are correlated to yield a score map and target is searched at different sizes (keep aspect ratio constant) in pyramid fashion during tracking. To better handle scale variations of the target, Li *et al.* extended this framework with region proposal network [51] proposed in Faster RCNN [69]. They employ multiple anchors with varying aspect ratios at each keypoint location in the search image. Score and shift of the anchors are regressed and the anchor with maximum score is picked during inference.

Although region proposal networks have a played a vital role in improving the performance of deep siamese trackers while maintaining real-time speed, enumerating multiple boxes at each keypoint location in the search region is still potentially inefficient and unsuitable for the task of single object tracking, where we just need to locate one target object. In this chapter, we suggest an alternate approach by directly regressing box offsets and sizes for keypoint locations in the search image. The proposed tracker, dubbed SiamReg, is fully convolutional, anchor-free and lighter in weight than the previous SiamRPN frameworks. We train our tracker end-to-end with Generalized IoU loss [70] for accurate bounding box regression and cross entropy loss for target classification. We perform several experiments on standard tracking benchmarks to demonstrate the effectiveness of our approach.

5.1 Method

Our framework consists of two submodules: feature extraction and bounding box regression. The feature extraction siamese module extracts relevant features of the tracking target in the current frame and template which are eventually correlated. The correlated features essentially gives the similarity between target-background and they are fed to our bounding box regression module. Our bounding box regression module yields a score and bounding box for each keypoint location in the search image. We first review the background on siamese tracking in brief, which will form the basis of our proposed tracker, SiamReg.

5.1.1 Siamese Framework for Tracking

In Siamese framework, the goal is to learn a robust similarity function which can find a template patch $\mathbf{z}$ in the search image $\mathbf{x}$. To achieve this, template patch is usually cropped from the first frame of a tracking sequence and search image is given from the following frame. Template and search images are then matched in the embedding space $\varphi(.)$ through a cross correlation operation:

$$f(\mathbf{z}, \mathbf{x}) = \varphi(\mathbf{z}) * \varphi(\mathbf{x})$$

The correlation operation yields a resulting scope map which is used to localize the target object in the search image. Note that the template and search branches share weights since we wish to match them in the same semantic space. For strict translation invariance, the backbone feature extraction network do not have any padding in the convolution layers [5]. Since we also require the spatial information to be retained in our fully convolution framework for accurate bounding box regression, we adopt this same siamese framework in our first stage.

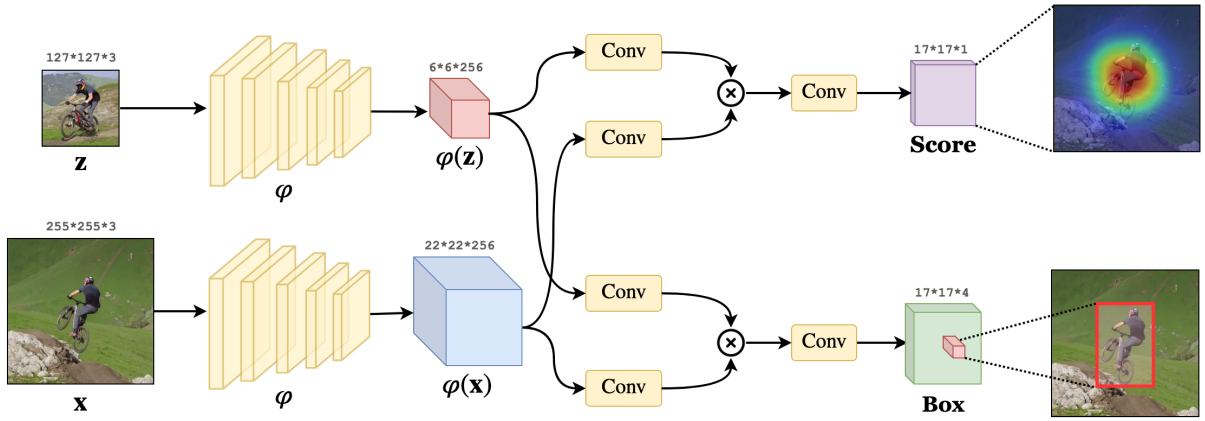

Figure 5.1 An overview of our proposed approach.

5.1.2 Fully Convolutional Bounding Box Regression

Let $I \in \mathcal{R}^{W \times H \times 3}$ be the search image of width W and height H. We wish to produce a score map $\hat{Y} \in [0,1]^{\frac{W}{s} \times \frac{H}{s} \times 1}$ and a size prediction map $\hat{S} \in \mathcal{R}^{\frac{W}{s} \times \frac{H}{s} \times 4}$, where s is the subsampling factor of our network. To obtain $\hat{Y}$ and $\hat{S}$, we convolve the template and search image features depth-wise and pass them through a few non-shared convolutional layers. We refer the reader to Figure 5.1 for the overall architecture of our tracker. Since our network is fully convolutional, each position in our output score map $\hat{Y}$ corresponds to a specific region in the search image. A prediction $\hat{Y} = 1$ suggests that the target object lies in that region whereas $\hat{Y} = 0$ indicates that it is background. Each position in size prediction map $\hat{S}$ yields 4 values corresponding to box offset and size: $\Delta x, \Delta y, w, h$. Thus, we get total $\frac{W}{s} \times \frac{H}{s}$ boxes in the search image.

Training: We obtain the groundtruth label maps for classification Y and regression S from the annotation box B^g. If the center of the annotation box lies inside a grid cell at position (m, n), we set $Y_{mn} = 1$ within a certain gaussian radius r centered at (m, n) else $Y_{mn} = 0$. Similarly, we build the regression map label S from the annotation box by calculating box offsets $\Delta x, \Delta y$ from the grid cell center. Box regression and score labels are clearly illustrated in Figure 5.2. Width w and height h are normalized with respect to the search image size and the offsets $\Delta x, \Delta y$ are normalized with respect to the grid cell size.

We train our classification branch with the standard logistic loss:

$$L_{cls} = \frac{1}{N} \sum_{mn} \begin{cases} \log(\hat{Y}_{mn}) & \text{if } Y_{mn} = 1 \\ \log(1 - \hat{Y}_{mn}) & \text{otherwise} \end{cases}$$

We retrieve the box B^p from the predicted regression map $\hat{S}$. In order to retrieve the box during training, we choose the grid cell location in the predicted regression map $\hat{S}$ where the center of the

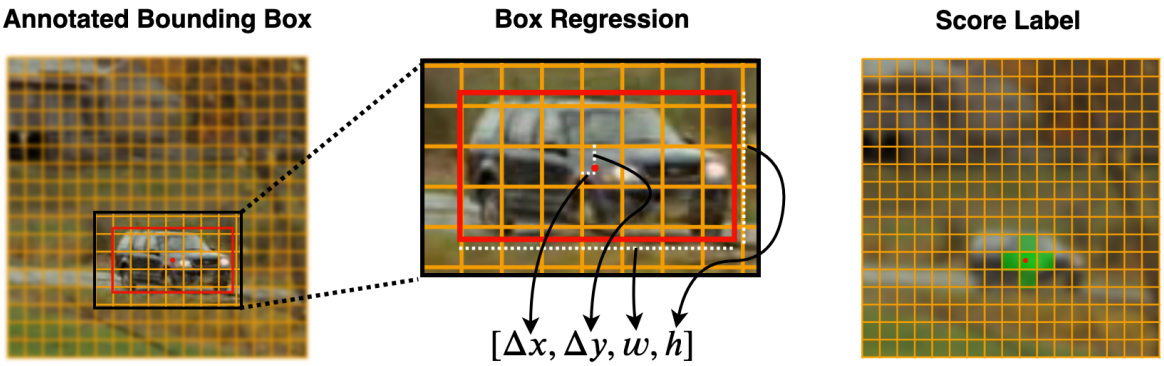

Figure 5.2 Illustration of SiamReg labels for target regression and classification.

groundtruth box B^g actually lies. We then simply calculate the GIoU loss [70] between the predicted box B^p and the groundtruth box B^g:

$$L_{reg} = 1 - GIoU(B^p, B^g)$$

When the boxes have non-zero overlap, GIoU behaves similar to the standard IoU metric. However, when the boxes are non-overlapping, GIoU scales proportionately to the distance between the two boxes. Optimizing the GIoU loss thus leads to better localization with time and it also indirectly optimizes our final evaluation metric IoU.

Finally, we optimize the following loss function:

$$L = L_{cls} + \lambda L_{reg}$$

where λ is a hyperparameter to balance the two losses for optimal training.

5.1.3 Tracking

We treat tracking as one-shot detection problem. During initialization, we store the template features and use it in the subsequent frames for depthwise correlation. In each frame, we get $\frac{W}{s} \times \frac{H}{s}$ boxes along with their corresponding scores after a forward pass through the architecture where W, H denotes width and height of search image and s denotes the subsampling factor of the network. Alike [51], we also apply the Hanning window and scale penalty on the score map. We obtain the predicted box by picking the one with maximum score. We update the size of tracking box using linear interpolation.

	VOT-2016			OTB-100		UAV-123		
	EAO ↑	Accuracy ↑	Robustness ↓	AUC ↑	Precision ↑	AUC ↑	Precision ↑	Speed
GOTURN [32]	-	-	-	0.427	0.572	0.451	0.702	100
KCF [33]	0.192	0.489	0.569	0.477	0.696	0.290	0.570	172
Staple [4]	0.295	0.544	0.378	0.578	0.784	0.453	0.697	80
SiamFC [5]	0.235	0.532	0.461	0.582	0.771	0.447	0.681	86
CFNet [82]	-	-	-	0.568	0.748	0.428	0.680	75
CREST [76]	0.283	0.514	1.083	0.623	0.838	0.396	0.649	1
MDNet [62]	0.257	0.541	0.337	**0.678**	**0.909**	0.464	0.725	1
SiamRPN [51]	0.344	0.560	0.260	0.637	0.851	0.527	0.748	200
SiamReg (Ours)	**0.367**	**0.615**	**0.238**	0.641	0.849	**0.575**	**0.764**	**204**

Table 5.1 Results on VOT-2016, OTB-2015 and UAV-123 tracking datasets.

5.2 Experiments

5.2.1 Implementation Details

Our tracker has been implemented in PyTorch with 2 Nvidia GeForce GTX 1080 GPUs. We use modified AlexNet [5] as backbone feature extractor and it is initialized with ImageNet before training. We freeze the first 2 layers and train the rest of the model on 4 datasets namely, COCO [55], ImageNet VID [71], YouTube Bounding Boxes [66], and ImageNet DET [71]. We adopt the same crop procedure as described in [51] to generate input image tuples. Our siamese framework takes 255×255 search image along with a 127×127 template image and produces 17×17 output maps. Our model is trained for 50 epochs and the learning rate is reduced in logarithmic fashion from 10^{-2} to 5×10^{-4}. During inference, we set the scale penalty and hanning window influence to 0.3.

5.2.2 Evaluation

Our tracker runs at 204 FPS. We test our tracker on 3 tracking benchmarks: VOT-2016 [24], OTB-100 [89] and UAV-123 [61]. We compare our tracker with the 8 state-of-the-art tracking algorithms: SiamRPN [51], MDNet [62], SiamFC [5], CFNet [82], GOTURN [32], CREST [76], Staple [4] and KCF [33]. SiamRPN, SiamFC and CFNet are based on the fully convolutional siamese framework discussed in Section 5.1.1. Note that we share the exact same backbone i.e. modified AlexNet as there in SiamRPN, SiamFC and CFNet to be fair in our comparison. GOTURN uses a pretrained AlexNet backbone with a few fully connected layers to directly regress the bounding box. CREST, Staple and KCF are popular correlation filtering based trackers. CREST uses deep residual features whereas Staple and KCF are based on handcrafted features.

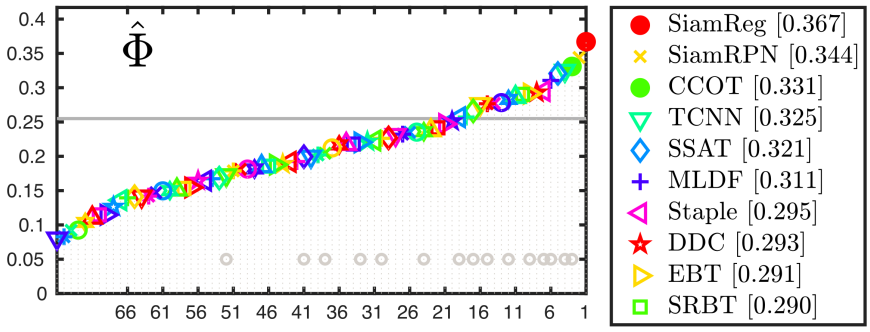

Figure 5.3 Overall VOT-2016 plot comparing with our proposed tracker SiamReg with 70 other trackers. The legend lists top 10 trackers on VOT-2016 sorted by Expected Average Overlap (EAO) metric.

5.2.3 Results

VOT-2016: VOT-2016 benchmark [24] borrows 60 sequences from VOT-2015 [46], however, the sequences are re-annotated with precise bounding boxes. As per VOT evaluation protocol, the tracker is restarted in the frame where the overlap between the tracker's prediction and groundtruth bounding box drops to 0. The trackers are ranked on the basis of three metrics: Expected Average Overlap (EAO), Accuracy and Robustness. Accuracy measures the average overlap between the prediction box and groundtruth box and Robustness gives the average number of tracking failures per sequence. Expected Average Overlap yields the mean overlap over the short-term subsequences. The results are summarized in Table 5.1. As evident from the Table 5.1, our tracker outperforms all the previous state of the art trackers on this benchmark including VOT-2015 challenge winner MDNet across the three evaluation metrics. Compared to SiamRPN which shares the same backbone, our proposed approach SiamReg brings a relative improvement of 9.8% in Accuracy and 8.4% in Robustness. Figure 5.3 illustrates the overall EAO plot of VOT-2016 benchmark, where SiamReg is plotted with 70 evaluated trackers. As per VOT-2016 report [24], all the trackers with EAO greater than 0.25 are considered to have state of the art performance.

OTB-100: OTB-100 [89] dataset consists of 100 sequences which pose various challenges like occlusion, illumination variation, motion blur, scale change etc. We follow the One Pass Evaluation (OPE) protocol in which the tracker is initialized in the first frame and there is no reset post failure. The trackers are evaluated with Area Under Curve (AUC) of success plots and precision metrics proposed in [87]. We present the results of SiamReg and other state of the art trackers in Table 5.1. On this benchmark, MDNet achieves the best performance on both the metrics. SiamReg improves upon SiamRPN in AUC and performs better than all the real-time trackers. The dominant performance of MDNet on this benchmark can be attributed to its online hard mining strategy, which effectively eliminates the distractors but it comes at the cost of tracking speed (1 FPS). On the other hand, our tracker SiamReg, do not update its model in online fashion, thus tracks at 204 FPS.

UAV-123: UAV-123 benchmark presents a unique challenge of tracking targets in videos captured from unmanned aerial vehicle. With this setup, the target is captured from multiple angles and altitude leading to variations in view point and scale. AUC under curve of success plots and precision at threshold of 20 pixels on UAV-123 dataset is reported in Table 5.1. SiamReg outperforms all the reported trackers with a large margin in AUC metric, specifically SiamRPN (0.527) and MDNet (0.464). We achieve significant improvement in overlap (AUC) and precision over SiamRPN which effectively highlights that our proposed anchor free bounding box regression framework is more effective in target scaling and localization than the anchor based SiamRPN.

5.3 Summary

In this chapter, we propose a novel and efficient siamese framework for visual object tracking. Our approach directly regresses bounding boxes in fully convolutional fashion. This avoids the use of anchor boxes which improves the target localization and scaling. Also, our architecture has fewer training parameters than the previous anchor based state of the art approach SiamRPN, which leads to faster inference and training. We show that our proposed approach SiamReg outperforms the previous baseline siamese frameworks as well as state of the art trackers on standard tracking benchmarks like VOT-2016, OTB-100 and UAV-123.

In future, we plan to improve the performance of the proposed approach by using stronger backbone architectures like ResNet [31] and DenseNet [36]. In the current setting, we simply used the exact same AlexNet backbone which is present in previous siamese trackers to show the effectiveness of our anchor free box regression module in a fair manner. Our qualitative analysis of the proposed approach also shows that it suffers from challenges like distractors and occlusions. More tracking cues could be utilized like optical flow and memory in the box regression module to handle these scenarios effectively instead of simply replying on the initial template frame. Lastly, a re-detection module is crucial in the long-term setting for the tracker to recover from failures. We leave these directions for future work.

Chapter 6

Conclusions

In this thesis, we addressed the long-term visual object tracking problem. To this end, we propose a large-scale TLP dataset in Chapter 3 to evaluate and study trackers from a long-term perspective. We show that our TLP dataset possess higher quality than existing tracking datasets in terms of scale, resolution and challenges. For thorough comparison betweeen short-term and long-term scenarios, we extracted a challenging short-term dataset, dubbed TinyTLP, from our TLP dataset. Our analysis shows that the performance of all the tracking algorithms drops by several folds from TinyTLP to TLP. To give a clear perspective, the performance of MDNet, which is one of the top trackers on OTB dataset [89] drops from 0.68 (TinyTLP) to 0.36 (TLP) in AUC metric. In trackers like MOSSE, the performance even drops by a factor of 10. One can always argue that the performance drop in long videos is just because of "more challenges or "frequent challenges in long-term. We conduct a small experiment to investigate this where we take a short sequence and repeat it 20 times to make a longer video out of it, by iteratively reversing and attaching it at the end to maintain the continuity. Our experiment results show that the tracker which performs well in the first repetition eventually fails in further repetitions, highlighting the accumulation in error or drift in tracking. We thus demonstrate that tracking performance not just depends on the challenges present in the sequence but also gets affected by its length.

In Chapter 4, we explore novel evaluation strategies to study the re-detection, recovery and reliability aspect in long-term tracking. Specifically, we test the re-detection capability of trackers *in the wild* with a novel setup: we simulate cuts virtually in TLP sequences by our GIoU minimization strategy. Our results show that all the trackers (both short-term and long-term) fail to recover on more than half of the sequences. The number of quick recoveries are even lesser for each tracker. Our recovery analysis quantitatively shows that trackers often recover in long-term by often tracking an alternate distractor object or the target passes over the tracker's location. State of the art trackers like SiamRPN [51] and ATOM [17] tracks an object other than the target for more than 13% of the time on average in a TLP sequence. Moreover, we indicate that trackers are reliable to use in long-term applications if the human effort to fix the incorrect tracker predictions is minimal. Keeping this in mind, we extend our proposed LSM metric for visual interpretation and present both qualitative and quantitative results.

Chapter 5 describes our novel siamese framework for object tracking. Although anchor based region proposal networks have significantly advanced the field of object detection and tracking, we show that better results can be achieved by a simple anchor free regression framework. Instead of estimating size by multiple anchors, we directly regress the grid offset and offset in the original image space. This not only consumes fewer parameters but also improves target localization and precision. We train our framework with GIoU loss for box regression and standard cross entropy for box classification. Preliminary experiments on standard tracking datasets like VOT-2016, OTB-100 and UAV-123 support our aforementioned claims. In the future, we plan to extend our proposed tracker with a long-term module which can enable efficiently recovering from tracking failures. Furthermore, multiple cues like optical flow, memory etc. could be utilized to make it robust to distractors and occlusions, which are the key challenges in object tracking.

Related Publications

1. Abhinav Moudgil and Vineet Gandhi. **Long-term Visual Object Tracking Benchmark**. *Asian Conference on Computer Vision (ACCV)*, 2018, Perth, Australia. (Oral)

2. Shyamgopal Karthik, Abhinav Moudgil and Vineet Gandhi. **Exploring 3 R's of Long-term Tracking: Re-detection, Recovery and Reliability**. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, Colorado, USA. (Under Review)

Bibliography

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

[2] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1619–1632, 2011.

[3] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*, pages 29–39. Springer, 2011.

[4] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, June 2016.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. *arXiv preprint arXiv:1606.09549*, 2016.

[6] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg. Unveiling the power of deep tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–498, 2018.

[7] J. Black, T. Ellis, P. Rosin, et al. A novel method for video tracking performance evaluation. *Proceedings of the IEEE InternationalWorkshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 03)*, pages 125–132, 2003.

[8] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[9] D. S. Bolme, B. A. Draper, and J. R. Beveridge. Average of synthetic exact filters. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2105–2112. IEEE, 2009.

[10] L. M. Brown, A. W. Senior, Y.-l. Tian, J. Connell, A. Hampapur, C.-F. Shu, H. Merkl, and M. Lu. Performance evaluation of surveillance systems under varying conditions. In *Proceedings of IEEE Pets Workshop*, pages 1–8. Citeseer, 2005.

[11] L. Cehovin, M. Kristan, and A. Leonardis. Is my new tracker really better than yours? In *Winter Conference on Applications of Computer Vision*, 2014.

[12] B. Chen, D. Wang, P. Li, S. Wang, and H. Lu. Real-time 'actor-critic' tracking. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[13] Z. Chen, Z. Hong, and D. Tao. An experimental survey on correlation filter-based tracking. *arXiv preprint arXiv:1509.05520*, 2015.

[14] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 6(4):271–288, 1998.

[15] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1631–1643, 2005.

[16] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[17] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Atom: Accurate tracking by overlap maximization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6638–6646, 2017.

[19] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.

[20] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[21] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[22] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[24] M. K. et al. The visual object tracking vot2016 challenge results. In *Proceedings, European Conference on Computer Vision (ECCV) workshops*, pages 777–823, 2016.

[25] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5374–5383, 2019.

[26] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. *arXiv:1703.05884*, 2017.

[27] V. Gandhi, R. Ronfard, and M. Gleicher. Multi-clip video editing from a single viewpoint. In *Proceedings of the 11th European Conference on Visual Media Production*, page 9. ACM, 2014.

[28] M. S. Grewal. *Kalman filtering*. Springer, 2011.

[29] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2011.

[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[32] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016.

[33] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015.

[34] Y. Hua, K. Alahari, and C. Schmid. Occlusion and motion reasoning for long-term tracking. In *ECCV*, 2014.

[35] L. Huang, X. Zhao, and K. Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*, 2018.

[36] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[37] C. Jaynes, S. Webb, R. Steele, and Q. Xiong. An open development environment for evaluation of video surveillance systems. *PETS02*, pages 32–39, 2002.

[38] I. Jung, J. Son, M. Baek, and B. Han. Real-time mdnet. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 83–98, 2018.

[39] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2011.

[40] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012.

[41] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1805–1819, 2005.

[42] H. Kiani Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *CVPR*, 2017.

[43] M. Kristan, S. Kovacic, A. Leonardis, and J. Pers. A two-stage dynamic model for visual tracking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(6):1505–1520, 2010.

[44] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[45] M. Kristan, J. Matas, A. Leonardis, et al. The visual object tracking vot2014 challenge results. In *ECCV Workshop*, 2014.

[46] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The visual object tracking vot2015 challenge results. In *ICCV workshops*, pages 1–23, 2015.

[47] M. Kristan, J. Perš, M. Perše, and S. Kovačič. Closed-world tracking of multiple interacting targets for indoor-sports applications. *Computer Vision and Image Understanding*, 113(5):598–611, 2009.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[49] M. Kumar, V. Gandhi, R. Ronfard, and M. Gleicher. Zooming on all actors: Automatic focus+ context split screen video generation. In *Computer Graphics Forum*, volume 36, pages 455–465. Wiley Online Library, 2017.

[50] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *arXiv preprint arXiv:1812.11703*, 2018.

[51] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[52] P. Li and H. Wang. Object tracking with particle filter using color information. In *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, pages 534–541. Springer, 2007.

[53] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV Workshops (2)*, pages 254–265, 2014.

[54] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24, 2015.

[55] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[56] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1704–1716, 2013.

[57] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[58] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, and M. Kristan. Now you see me: evaluating performance in long-term visual tracking. *arXiv:1804.07056*, 2018.

[59] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *CVPR*, 2015.

[60] A. Moudgil and V. Gandhi. Long-term visual object tracking benchmark. *arXiv preprint arXiv:1712.01358*, 2017.

[61] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016.

[62] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[63] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

[64] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[65] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical recipes in c++. *The art of scientific computing*, 2:1002, 1992.

[66] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017.

[67] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[68] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[69] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[70] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[71] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[72] O. Russakovsky, L.-J. Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2121–2131, 2015.

[73] H. Smail, H. David, and S. D. Larry. W4: Real-time surveillance of people and their activities. *IEEE transactions on pattern analysis and machine intelligence*, 22(8), 2000.

[74] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *TPAMI*, 36(7):1442–1468, 2014.

[75] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang. Crest: Convolutional residual learning for visual tracking. In *ICCV*, 2017.

[76] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M.-H. Yang. Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2555–2564, 2017.

[77] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. Lau, and M.-H. Yang. Vital: Visual tracking via adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8990–8999, 2018.

[78] J. S. Supancic and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013.

[79] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in neural information processing systems*, pages 2553–2561, 2013.

[80] C. Tomasi and T. K. Detection. Tracking of point features. Technical report, Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, 1991.

[81] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2805–2813, 2017.

[82] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2805–2813, 2017.

[83] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. Smeulders, P. Torr, and E. Gavves. Long-term tracking in the wild: A benchmark. *arXiv:1803.09502*, 2018.

[84] P. Viola, M. Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, 1(511-518):3, 2001.

[85] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, pages 1–21, 2012.

[86] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, Sept. 2015.

[87] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

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

[89] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[90] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[91] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2711–2720, 2017.

[92] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, 2017.

[93] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014.

[94] T. Zhang, C. Xu, and M.-H. Yang. Multi-task correlation particle filter for robust object tracking. In *CVPR*, 2017.

[95] L. Cehovin Zajc, A. Lukezic, A. Leonardis, and M. Kristan. Beyond standard benchmarks: Parameterizing performance evaluation in visual object tracking. 2017.