Tampere University

UĞUR KART

Visual Object Tracking on RGBD Videos Using Discriminative Correlation Filters

Tampere University Dissertations 452

Tampere University Dissertations 452

UĞUR KART

Visual Object Tracking on RGBD Videos Using Discriminative Correlation Filters

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty of Information Technology and Communication Sciences
of Tampere University,
for public discussion in the auditorium TB109
of Tietotalo, Korkeakoulunkatu 1, Tampere,
on 8 October 2021, at 12 o'clock.

ACADEMIC DISSERTATION
Tampere University, Faculty of Information Technology and Communication Sciences
Finland

Responsible supervisor and Custos	Professor Joni Kämäräinen Tampere University Finland	
Supervisor	Professor Jiří Matas Czech Technical University Czech Republic	
Pre-examiners	Professor Michael Felsberg Linköping University Sweden	Assoc. Professor Dirk Kraft University of Southern Denmark Denmark
Opponent	Assoc. Professor Ville Kyrki Aalto University Finland	

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Cover design: Roihu Inc.

ISBN 978-952-03-2057-7 (print)
ISBN 978-952-03-2058-4 (pdf)
ISSN 2489-9860 (print)
ISSN 2490-0028 (pdf)
http://urn.fi/URN:ISBN:978-952-03-2058-4

PunaMusta Oy – Yliopistopaino
Joensuu 2021

Dedicated to my parents Mehmet and Perihan, without whom none of this would be possible.

PREFACE/ACKNOWLEDGEMENTS

This study has been carried out at Tampere University, Finland during the years 2017 - 2019.
Firstly, I would like to express my gratitude to my supervisor Professor Joni Kämäräinen for his endless support at every step of my thesis journey. Secondly, I would like to thank Professor Jiří Matas for his wonderful guidance. He taught me paying attention even to the smallest details can make a huge difference.
I am also thankful to my co-authors, especially to Alan Lukežič and Matej Kristan for all the insightful discussions and help.
Finally, I am forever grateful to my family for their lifelong, unconditional support and love.

Tampere
31.01.2021
Uğur Kart

ABSTRACT

The amount of digital media created annually has recently increased exponentially due to the ubiquitousness of digital cameras. This increase has also made manual analysis of the data impossible, hence, computer vision algorithms have been adopted to automatically analyze and extract meaningful information. One particular field of computer vision is generic visual object tracking, where the location of the target is only known in the first frame and the algorithm tries to detect its location in the following frames. Most of the existing literature has focused on tracking using traditional RGB cameras however, lately, cheap and reliable depth sensors have become easily accessible. This has opened up new possibilities for many computer vision algorithms including object tracking.

Depth information allows algorithms to leverage 3D understanding of the surroundings. Adding this new dimension can be particularly helpful in tracking where occlusions and shape changes are common. This thesis, focuses on generic visual object tracking on RGBD cameras and contributes to the field in two different ways. Firstly, it develops novel object tracking algorithms using RGBD data. For achieving this, it builds on top of elegant yet powerful fundamentals of spatially constrained correlation filters. To the best of our knowledge, this is the first attempt to use depth inherently in a correlation filters framework. Additionally, it also takes advantage of depth data for detecting self and external occlusions. This allows dynamic adjustment of filter updates to avoid model corruption. Evaluations on publicly available benchmarks validate the proposed algorithms by showing that they achieve state of the art results.

Second contribution of this thesis is the creation of a novel RGBD tracking dataset. This dataset addresses the shortcomings of previous RGBD benchmarks and provides a new challenge to facilitate new RGBD trackers.

TIIVISTELMÄ

Digitaalisen median määrä ja käyttö on kasvanut eksponentiaalisesti johtuen digitaalisten kameroiden ja kamerapuhelimien määrän kasvusta. Suuresta määrästä johtuen kuvien ja videoiden manuaalinen käsittely ja analysointi ei enää ole mahdollista. Automaattista käsittelyä ja analyysia varten kehitetään jatkuvasti parempia tietokonenäköalgoritmeja. Yksi tärkeä analyysiongelma on visuaalinen kohteiden seuraaminen, jossa seurattava kohde on merkitty vain ensimmäiseen videon ruutuun ja menetelmän pitää pystyä seuraamaan sitä läpi videon. Kohteiden seuraamista on tutkittu laajasti ja hyviä menetelmiä löytyy erityisesti perinteisten värikameroiden (RGB-kamera) tuottamille videoille. Viime aikoina ovat kuitenkin myös RGBD-kamerat yleistyneet ja näissä on värikuvan lisäksi myös syvyystieto (etäisyystieto) käytettävissä. Syvyystieto auttaa mallintamaan kohteen liikettä 3D-ympäristössä ja tekee esim. kohteiden peittymisen (eng. "occlusion") ja 3D-muodonmuutosten havaitsemisen helpommaksi.

Tämä väitöskirjatyö käsittelee visuaalista kohteenseurantaa RGBD-kameroiden tuottamalle videolle ja tekee kaksi merkittävää tieteellistä kontribuutiota. Ensimmäisenä kontribuutiona työ esittelee uuden RGBD-menetelmän kohteenseurantaan. Menetelmä pohjautuu spatiaalisesti rajoitettuun korrelaatiosuotimeen, joka on ollut menestyksekäs menetelmä RGB-kohteenseurannassa. Uusi menetelmä on kuitenkin ensimmäinen, jossa syvyystieto on yhdistetty korrelaatiosuotimiin. Menetelmä hyödyntää syvyysdataa kohteen peittymisen havaitsemisessa. Peittymisen havaitseminen mahdollistaa kohteen mallin päivittämisen vain niiltä osin kuin kohde on näkyvissä. Esitelty menetelmä saavuttaa parhaat kirjallisuudessa tunnetut tulokset julkisilla testaustietokannoilla. Väitöskirjan toinen tärkeä kontribuutio on uusi testaustietokanta RGBD-kohteenseurantaan. Uusi tietokanta paikkaa aikaisemmissa tietokannoissa olleita puutteita ja näin mahdollistaa entistä parempien menetelmien kehittämisen.

CONTENTS

ORIGINAL PUBLICATIONS

Publication I U. Kart, J.-K. Kämäräinen, J. Matas, L. Fan and F. Cricri. Depth Masked Discriminative Correlation Filter. (2018).

Publication II U. Kart, J.-K. Kämäräinen and J. Matas. How to Make an RGBD Tracker ?: (2018).

Publication III U.Kart, A. Lukežič, M. Kristan, J.-K. Kämäräinen and J. Matas. Object Tracking by Reconstruction with View-Specific Discriminative Correlation Filters. (2019).

Publication IV A. Lukežič, U. Kart, J. Käpylä, A. Durmush, J.-K. Kämäräinen, J. Matas and M. Kristan. CDTB: A Color and Depth Visual Object Tracking Dataset and Benchmark. (2019).

Author's contribution

Publication I In [P1], depth maps are explored for mask generation in correlation filters framework. The author implemented the proposed method, performed the tests and wrote the paper using the valuable insights from co-authors and the supervisor.

Publication II In [P2], color and depth are used as cues for mask generation in correlation filters framework with a faster re-detection algorithm. The author implemented the proposed method, performed the tests and wrote the paper using the valuable insights from co-authors and the supervisor.

Publication III In [P3], 3D structure of the object is employed for out-of-plane rotation detection. The author implemented the proposed method, performed the tests and wrote the paper using the valuable in-

sights from co-authors and the supervisor.

Publication IV — In [P4], a novel RGBD dataset is collected using multiple modalities. The author helped collection and the annotation of the data, performing the experiments and paper writing process.

1 INTRODUCTION

Recent decades have seen an exponentially grown amount of digital media thanks to the availability of the affordable devices with video recording capabilities. However, this enormous increase also came with a price which is the difficulty of analyzing the data automatically within a reasonable period of time. To address this problem, many different computer vision algorithms have been developed.

Visual Object Tracking (VOT) is one of the fundamental problems in computer vision where the aim is to automatically localize the target object whose location is "only" known in the first frame of a given video sequence as illustrated in Fig. 1.1.

Due to its intuitive interpretation, it has numerous applications in real-life scenarios. For example, sports [75], robotics [25], military [21], surveillance [128] among others. Therefore, it has taken the attention of the computer vision community from an early stage and has been researched in depth since then [62, 64, 65, 66, 87, 88, 112, 133].

The earliest attempts to solve this problem employed relatively simple approaches. In general, they assumed minimal displacement between image frames and relatively constant illumination [77]. Although these methods worked to a certain extent, they were insufficient to provide a reliable solution in the "wild" where challenging appearance changes and scenes are common. With the advent of the powerful processors thanks to the developments in semiconductor technologies, more complex methods have been adopted to successfully solve this problem [31, 32, 80, 81]. However, there is still plenty of room to improve.

Despite the excellent results obtained in the public benchmarks [62, 63], the vast majority of the VOT literature still focuses on the RGB channels. This can be attributed to the fact that until recently, depth acquiring sensors were not affordable and widespread. However, within the last decade, this situation has changed drastically thanks to the cheap and good quality depth sensors. Nowadays, it is even possible to find depth sensors in mobile phones such as iPhone X, Huawei P30 pro

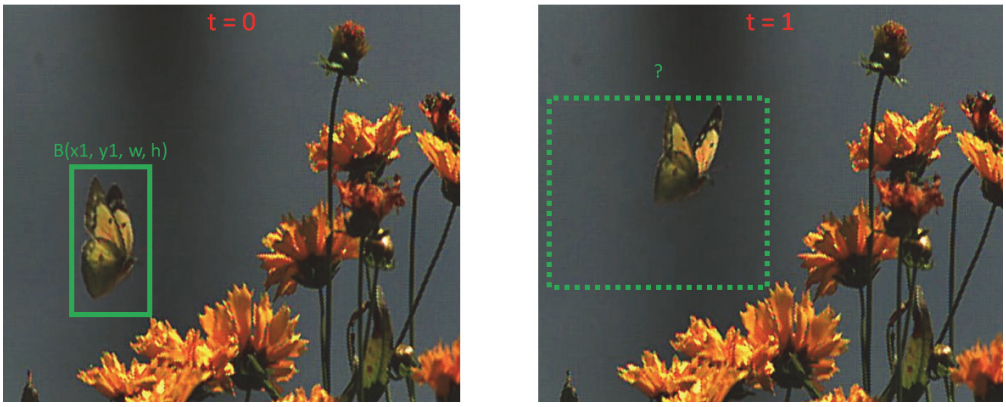

Figure 1.1 Aim of Visual Object Tracking. The target location is known only in the first frame and the VOT algorithm tries to locate it in the following frames.

etc. This reality brings in new opportunities to VOT by adopting depth channel as a fourth information channel in addition to the RGB channels when designing novel algorithms.

Depth is an important information source that helps human beings understand the 3D world around us. Hence, it is natural to include it in the computer vision algorithms where we try to bring this understanding to the computers. In VOT, depth can be used for detecting occlusions, out-of-plane rotations and also to have more discriminative features by leveraging the 3D structure of the target object and its background.

In this thesis, we propose novel VOT algorithms which use depth channel as a fundamental information source to address some of the problems mentioned above. We claim and experimentally support that depth information provides significant gains when added to RGB trackers and it enables new research opportunities to explore even more accurate tracking performance.

The objectives of this thesis are given as;

- To conduct a comprehensive literature survey on RGB and RGBD trackers
- To propose novel algorithms for VOT on RGBD sequences
- To clarify the importance of depth in VOT by comparing depth augmented trackers against their RGB baselines
- To propose a novel and extensive RGBD tracking benchmark for facilitating further developments in the field

The remainder of this thesis is organized as follows; in Chapter 2, an in-depth review of the existing VOT modalities, algorithms and their performance measures are provided. Chapter 3 presents the historical development and the mathematical foundations of the tracking paradigm adopted in this thesis. Chapter 4 offers the contributions of the author to this thesis and finally Chapter 5 concludes the thesis and makes the final remarks.

2 VISUAL OBJECT TRACKING

In this chapter, an in-depth review of the existing VOT literature will be provided with trackers divided into families according to their common properties. However, please note that a tracker can be included in multiple families since it can have features from different classes. For example, a tracker can be included in both short-term trackers and generative trackers. Since the majority of the literature is about single object, single target, RGB trackers, we found it suitable to provide separate sections for multi-object trackers and RGBD trackers.

Our taxonomy consists of tracking duration in Section 2.1, multi-object trackers in Section 2.2 representation of the target object in Section 2.3, usage of depth information in Section. 2.4. Following the taxonomy, commonly used datasets and the performance measures are also introduced in Section 2.5.

2.1 Trackers According to the Duration of Tracking

There are two main groups in terms of tracking duration in the field of VOT. First group consists of short-term trackers which do not have any target loss detection and re-detection capabilities. This family of trackers are initialized with a bounding box on the first frame and they try to localize the target object in the rest of the sequence without knowing whether the tracker has drifted or not. On the other hand, the second group focuses on trackers that can detect the target object loss and take action accordingly *e.g.* stopping model updates and target re-detection [62]. In this chapter, these two families of trackers will be examined in detail.

2.1.1 Short-Term Trackers

Short-term object tracking is the most commonly studied family of trackers as it creates the basis for VOT. In general, the short term trackers are initialized on the

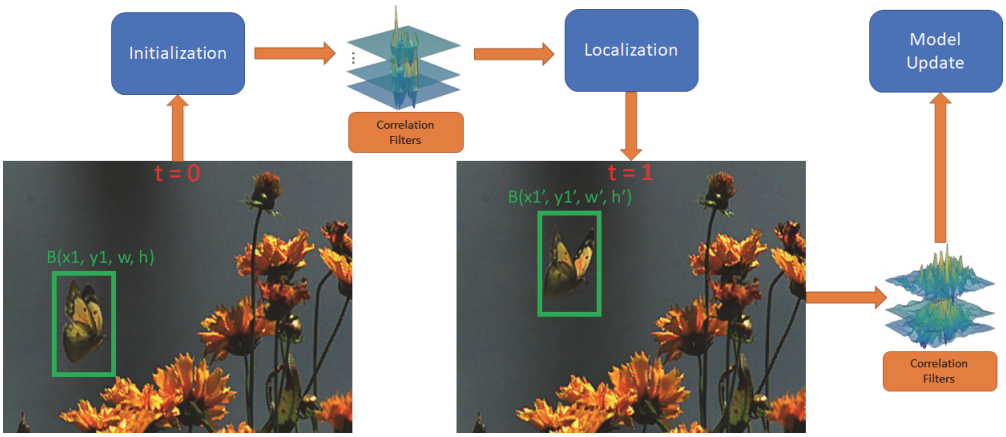

Figure 2.1 Short Term Tracking Process With Correlation Filters ©2017 IEEE

first frame with an axis-aligned bounding box and they try to localize the target object in the next frames. To achieve this, three main building blocks are commonly adopted in all trackers although of their inner workings can differ; *initialization*, *localization* and *model update*. This process is illustrated below in Fig. 2.1.

The initialization stage is when the target object is represented in terms of extracted features such as pixel values [13], HOG features [80] or recently deep features [31, 32]. These extracted features are later used to identify the target object location in the other frames. After the initialization is completed, the tracker is fed with a new input frame where the target needs to be localized. The tracker searches for similar features in the vicinity of the previous location since the inter-frame location displacement is assumed to be minimal. Once the target is found, a new set of features are extracted from the new location and the target model is updated with the new feature set to maintain an up-to-date target appearance.

One of the earliest methods to tackle short-term tracking problem is Lucas-Kanade tracker [77]. In this seminal paper, the authors propose to localize the target object by adopting an affine transformation matching scheme. Lucas *et al*. take advantage of spatio-temporal derivatives to cope with the changes in scale, rotation and translation. Another template matching based algorithm was proposed by Briechle *et al*. [18] in a normalized cross-correlation framework where a direct matching between the target object and candidate regions is performed. Region with the maximum matching score is considered as the new target position. Comaniciu *et al*. proposed a mean shift tracking algorithm [27] in which the target is represented with

an RGB color histogram and the matching is performed using Bhattacharrya metric. Adoption of histogram provides immunity against rotation. However, it also makes it vulnerable against illumination changes.

An important development in short-term tracking was the introduction of discriminative correlation filters (DCF) which was pioneered by Bolme *et al*. [13]. The authors proposed to use circular correlation in Fourier Domain where the correlation can be performed as a term-by-term multiplication between two matrices thanks to the properties of Fourier Transform. In this paper, Bolme *et al*. train DCF by minimizing the sum of squared error between the actual correlation error and the desired target which is usually a 2D Gaussian distribution. The localization is then performed by simply applying correlation in Fourier Domain between the region of interest (ROI) and the trained filters. Once the target is localized, the model is updated as a weighted average of the previous and the current filters.

This simple, efficient yet very successful method captured the attention of the visual object tracking community and a plethora of trackers have been proposed by capitalizing on the correlation filters since then. Henriques *et al*. [48] analytically showed the relationship between regularized correlation filters and the circulant matrices. This derivation provided an important insight to the success of the method proposed by Bolme *et al*. [13] as by their nature, circulant matrices implicitly create significant amount of training samples by circular shifts. Moreover, Henriques *et al*. also showed that it is possible to adopt more powerful non-linear filters using the *"kernel trick"* similar as in Support Vector Machine(SVM) [28]. Danelljan *et al*. [30] first extended correlation filter tracking to be scale-adaptive to tackle target size changes and later proposed a method to tackle features with different resolutions via an implicit interpolation [32]. As a further improvement for [32], Danelljan *et al*. proposed an efficient convolution framework where the authors addressed the speed and efficiency problems arose from the adoption of deep features by using creating a small set of base filters which are then linearly combined for creating feature layers [31].

A well-know issue in correlation filter tracking is the boundary effects as illustrated in Fig. 2.2 .This problem stems from the implicit creation of the spatially shifted training samples by the circular correlation operation which then causes unrealistic negative training samples. Galoogahi *et al*. tackled this problem by adding an augmented spatial objective to the correlation filter formula [40]. In practice,

Figure 2.2 Boundary effects as a result of canonical correlation operation. Green boxes represent the positive and red boxes represent the negative implicit training samples respectively [40] ©2015 IEEE

this meant adding a very large binary mask that represents the active parts of the signal to the filter training formula. However, this addition also imposed the constraint that the problem is needed to be solved in the spatial domain and hence, the complexity changed from $\mathcal{O}(NDlogD)$ to $\mathcal{O}(D^3 + ND^2)$ because the usual closed-form solution for the correlation tracker training is no longer valid. To circumvent this, Galoogahi *et al*. posed this as an optimization problem by adopting an Augmented Lagrangian Framework [16]. Despite its novelty, the method proposed by Galoogahi *et al*. still represented the spatial support as an axis-aligned bounding box. Recently, Lukezic *et al*. adopted a graph-cut based approach to precisely estimate the spatial support for the target object [80]. Instead of using an axis-aligned bounding box, the authors created foreground masks by using a per-pixel segmentation approach.

2.1.2 Long-Term Trackers

Long-term tracking can be considered as a superset of short-term tracking. However, its literature is far more limited [62]. In order to be a long-term tracker, the method should have either an implicit or explicit awareness of the tracking quality. By using this measure, the tracker then can decide whether to stop updating its appearance model and/or enter into a re-detection state.

Pernici *et al*. [103] adopted a Scale Invariant Feature Transform (SIFT) [74] based bag-of-words [111] approach for representing the local object appearance along with a non-parametric learning method that avoids target model updates in case of occlusions. Grabner *et al*. [41] proposed an online, semi-supervised boosting approach to avoid target model corruption while updating. Chang *et al*. [23] proposed to imitate human visual system to detect tracker failures and hence, stopping tracker model updates to prevent drifting. Kwak *et al*. [116] divided the target into grids and produced a per-grid occlusion probability map which is then used for deciding whether to fallback onto a motion based tracking.

Pioneered by Kalal *et al*. [55], a three-component, long-term tracker paradigm has started to gain attraction recently. These components can be named as; a short-term tracker for target localization under normal circumstances, a target detector to re-detect the target object in case of its disappearance/reappearance and a third block that decides which component (short-term tracker or target detector) to be used at any given frame [55, 62, 81]. An overview of this family of trackers' internal structure can be found in Figure. 2.3 below.

Kalal *et al*. [55] used a *p-n* learning scheme where *p expert* detects the false negatives and *n expert* detects the false positives. Ma *et al*. [84] proposed the usage of random fern classifier [102] as the target re-detector and it is enabled in case of a failure detection which is determined by analyzing the response score from the correlation filter. The fern classifier is then applied in a sliding window manner on the whole image plane. As a further expansion to their work, Ma *et al*. [83] adopted a multi-correlation filter tracker based approach where the authors used one tracker with large filter update rate for taking inter-frame appearance changes into account, one tracker for scale change detection and one tracker with small filter update rate to create a long-term appearance memory. By analyzing the response scores obtained from this long-term memory, tracker failures are detected. Hong *et al*. [50] used

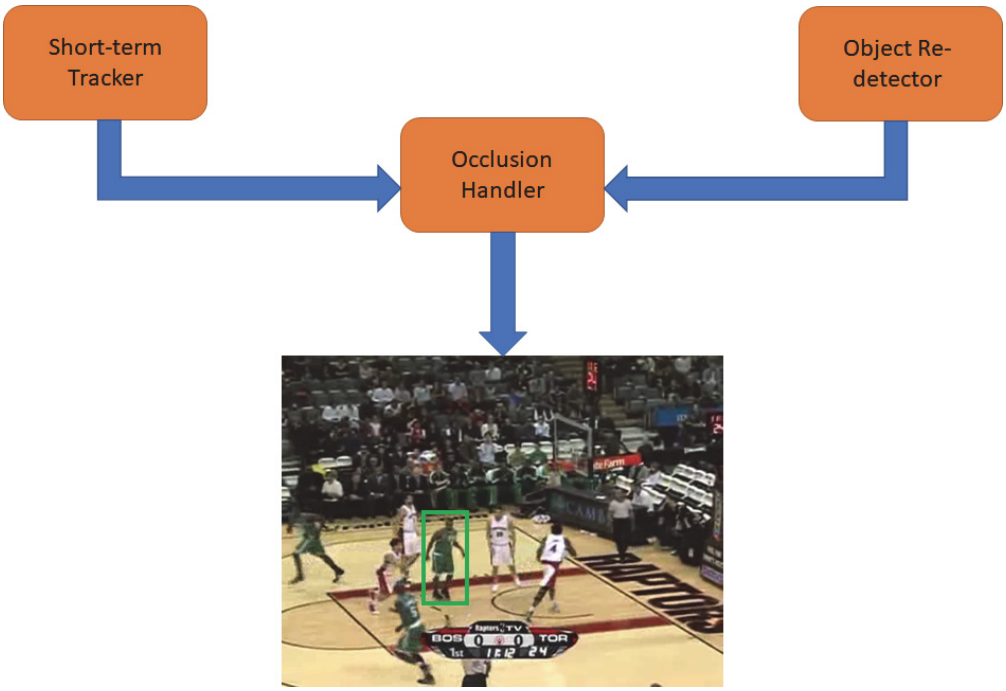

Figure 2.3 Building blocks of long-term tracker structure prposed by Kalal *et al.* [55]

cognitive psychology principles by proposing a short-term and long-term memory based tracker. An integrated correlation filter tracker is adopted for short-term localization while a keypoint matching and RANSAC [120] based algorithm used for keeping track of long-term changes. They detect the occlusion by considering the keypoints that belong to the background but inside the target bounding box. This detection is used for deciding whether to update the appearance model or not. Fan *et al.* [35] proposed a correlation filter tracker based method supervised by a DNN to verify and if needed, correct the tracker results. As an another correlation filter based long-term tracker, Lukezic *et al.* [81] proposed a fully-correlational approach. The tracker is built upon a short-term correlation tracker and a long-term correlation detector. The long-term component stores multiple correlation filters that are updated at different temporal rates. In case of target loss, an adaptive search region growth is also adopted to take target displacement into account.

2.2 Multi-Object Trackers

Another important classification for the tracker algorithms is their taxonomy according to the number of simultaneously tracked objects. So far, the trackers we have presented have all been designed to track a single target. In this section, we will explore the multi-object tracker in which the goal is matching the observations (*e.g.* detections) to the target states in an image sequence usually solved as a probabilistic estimation problem [82]. Luo *et al.* [82] divide the aim of MOT into three main parts; localization of the target objects, avoiding target identity swaps and tracking the target objects successfully between frames in presence of frequent occlusions.

Although there are many sub-categories for MOT algorithms, we will present two main families in scope of this thesis which are *online* and *batch* trackers. *Online* trackers process frames sequentially as it is received and has no information about the future frames whereas *batch* trackers process multiple frames (batches) at once.

We invite the reader to read [82] for a more detailed analysis of MOT literature.

2.2.1 Online Trackers

Online trackers can be considered similar to the single-object trackers in terms of how the information is processed. On every frame, the tracker tries to match the current states of the target objects to the observations extracted from the frame and update its model. Hence, this family of trackers is more suitable for certain real-life applications such as autonomous driving.

Kim *et al.* [60] proposed a multiple hypothesis approach and introduced a delayed data association mechanism where numerous hypotheses are stored for each track until they are pruned as a result of divergence from a global hypothesis. Naiel *et al.* [96] adopted a flock of single object trackers coupled with a motion model within a particle filter framework. Bae *et al.* [4] used tracklet confidence scores for tracklet association. Xiang *et al.* [134] modelled the objects with a Markov Decision Process [5]. Ristani *et al.* [105] proposed a real-time algorithm in which the MOT problem is posed as a graph partitioning problem that is solved in a two-stage cascade manner for computational complexity concerns. Zhu *et al.* [139] employed a short-term tracker and used it along with an object detector to continuously monitor the possible track losses. Bergmann *et al.* [7] adopted only an object detector based on Faster

R-CNN [104] to use as the tracker. The authors proposed a two-level pipeline to regress object trajectories and kill/start tracks. Voigtlaender *et al*. [125] solves MOT problem jointly within a detection, tracking and segmentation framework by extending Mask R-CNN [45] into temporal domain.

2.2.2 Batch Trackers

Contrary to the online trackers, batch trackers fetch previous and future frames to construct an optimal trajectory for each object. Zamir *et al*. [106] globally associates appearance and motion using the full temporal span by adopting a generalized minimum clique graphs framework. Berclaz *et al*. [6] divides the scene into discrete grids and calculates a probabilistic occupancy map for each grid using an object detector. Per frame individual detections are then connected using a K-Shortest Path algorithm [117]. Chari *et al*. [24] added pairwise costs and solved the tracking problem within a min-cost network flow framework [69]. Milan *et al*. [93] defined a global continuous energy function on the whole sequence by taking into account not only appearance cues but also the physical limitations for every object available. This energy function is then optimized by alternating between conjugate gradient descent [49] and five different discontinuous jumps. Wen *et al*. [129] proposed to use an undirected hierarchical relation hypergraph based method in which the tracking is reduced to a search for the optimal candidate in multiple dense neighboring nodes [71]. Feichtenhofer *et al*. [36] extended an object detector [29] to train an end-to-end fully convolutional deep neural network (DNN) for jointly solving detection and tracking problem. The authors adopted ResNet-101 [46] to extract convolutional features which are used for calculating the cross-correlation between the adjacent frames. Maksai *et al*. [85] were first to propose a non-Markovian global optimization to take behavioral patterns into account to further improve the traditional, tracklet-based MOTs. Wang *et al*. [126] tackled the suboptimality of the generic min-cost network usage in MOT algorithms by proposing a Minimum Update Successive Shortest Path which provides an exact, optimal solution.

2.3 Trackers According to the Representation of the Targets

In terms how the target objects are represented, the trackers can be grouped under two families namely *generative* and *discriminative*. Generative trackers creates an appearance model of the target object and try to match this model in the next frame (*e.g.* template matching) whereas discriminative trackers continuously train a classifier to be evaluated on the candidate locations to separate foreground from background.

2.3.1 Generative Trackers

Briechle *et al.*'s Normalized Cross Correlation [18], Comaniciu *et al.* [27] and Lukas-Kanade's Lukas-Kanade Tracker [77] can be considered few of the earlier works belonging to this family of trackers. Nguyen *et al.* [99] proposed an adaptive appearance based method in which a 20 x 20 template is used to represent the target. Each value in this template is linked to a separate Kalman Filter [56] with shared parameters to produce temporally smoothed results. Rivlin *et al.* [1] proposed a part based method similar to bag-of-words [111] where the comparison between the patches are made using histogram and the results of each patch are aggregated. Oron *et al.* [101] established a similarity metric by jointly considering the pixels in spatial and appearance domains. The authors calculated the similarity between two patches by adopting the Earth Moving Distance [108]. Ross *et al.* [107] adopted an adaptive approach using Principal Component Analysis [119] by extracting eigenvalues of the target template and kept these values in a shifting window manner. This allowed tracker to store a larger variation of target appearances spread over a long time period. Kwon *et al.* [67] proposed a method based on particle filter framework on 2-D affine transformation group and a Gaussian sampling. In another work from Kwon *et al.* [54], a flock of trackers method is adopted where the number of trackers is dynamically changed depending on the target status (*e.g.* occlusions or strong appearance changes). The authors proposed to not only sample the target status but also the trackers themselves. Cehovin *et al.* [22] tackled fast appearance changes by coupling global and local visual cues in a part based manner. While the local layer provides fast adaptation to the changes in the target appearance and geometry, the global level ensures the consistency of the addition/removal of local patches by imposing color, motion and shape constraints. Mei *et al.* [89] sparsely represented the

target as a linearly combined set of templates constructed from a set of base templates via $L1$ minimization [130] and adopted a particle filter for sampling templates. In order to handle the occlusions, they proposed to use trivial templates where there is only one non-zero element. To further improve the efficiency, Mei *et al*. [90] proposed to use an early pruning method to alleviate the computational complexity issues arose from $L1$ minimization. In this work, the authors also proposed an explicit occlusion detection mechanism to avoid leaking the trivial templates into the template set.

2.3.2 Discriminative Trackers

Discriminative trackers pose the tracking problem as a foreground/background classification. To this end, they train a classifier using the previously observed measurements and evaluate it on the new frame to localize the target. Due to their proven success, this family of trackers have seen a surge in recent years. For example, 76% and 79% of the short-term trackers submitted to Visual Object Tracking Challenge 2018 and 2019 were from this family [62, 63].

Hare *et al*. [44] proposed a kernelized structured output SVM [12, 122] based method and treated the tracking problem as regression rather than classification. This allowed to alleviate the problems in the traditional discriminative trackers related to the binary labelling such as the equally weighted training samples regardless of their positions. The prediction function is then solved by minimizing a convex objective function using [14, 15]. Instead of solving a single ridge regression as common in most DCF trackers, Bertinetto *et al*. [9] proposed to train two templates; one with HOG [95] and one with color histograms and then, linearly combine the results of the two for two main reasons. First is the fact that HOG and color histograms are complementary features since HOG represents the shape and robust against illumination. However, it is also vulnerable to deformations. Color histograms on the other hand are susceptible to changes in the illumination however, they are invariant to deformations. The immense success of DNN in computer vision [110] has naturally caught the attention of the researchers working on tracking as well. Held *et al*. [47] trained an offline classifier by optimizing over the inter-frame displacement on successive frames using $L1$ loss. Tracking is performed by first applying convolutional layers on both the previous and the current frame which are

then connected to a fully-connected layer to predict the target location. Bertinetto *et al*. [10] interpreted the tracking problem as a similarity learning problem and proposed to use a Siamese DNN architecture [19, 118, 136]. The authors trained a network on ILSVRC15 [110] which is used for extracting features from the previous and the current frames. These features are then correlated to obtain the target location. Valmadre *et al*. [124] further improved the work from Bertinetto *et al*. [10] and provided a closed-form solution to the correlation filter to be inherently included in the learning process of a Siamese DNN architecture. Nam *et al*. [97] trained a convolutional neural network (CNN) with multiple of fully-connected, final layers which share common initial layers (three convolutional and two fully connected). At each iteration, only a single final layer is enabled to encode the sequence specific information along with the common layers to encode the inter-sequence information. In test time, only the last three, fully connected layers are updated to robustly adapt into new target states.

In addition to the methods mentioned in this chapter, all previously mentioned, DCF based trackers [13, 31, 32, 39, 40, 48, 80] and TLD [55] belong to this group as well.

2.4 RGBD Trackers

In this section, we will examine the literature on RGBD trackers under *Generative Trackers* and *Discriminative Trackers* since they are all single object, long-term trackers. Hence, a taxonomy under the target representation is deemed to be sufficient.

2.4.1 Generative Trackers

With the advent of Microsoft Kinect v1, there has been a significant increase in the number of RGBD trackers in the literature as it has become significantly easier to capture video sequences with precise depth data. As one of the pioneers in the field, Song *et al*. [113] proposed a novel RGBD dataset and nine baseline algorithms by combining different approaches such as 2D optical flow [20], 3D iterative closest point [138], HOG [95], color names [127], 3D shapes [53] and the number of points in a voxel. Among these nine methods, the one which uses RGBD data + optical flow + occlusion handling where the occlusion handling is achieved by modelling

the depth data with a 1-D Gaussian obtained the top performance. A new rising peak with a smaller depth data than the target object is considered as a potential occluder. Despite its novelty and promising accuracy, the reported speed 0.26 FPS made this method infeasible for real-life applications.

Liu *et al*. [72] extended the original mean-shift tracker [27] into a 3D mean-shift algorithm with explicit occlusion handling capabilities using a multitude of heuristics. Instead of the 2D bounding box in the classical meanshift tracker [27], the authors proposed to use a 3D bounding sphere to construct the color histogram. Each point in this sphere then contributed to the histogram according to its 3D euclidean distance.

Meshgi *et al*. [92] used a particle filter framework where they represented the occlusions with a latent variable with which they take actions such as expanding the search region to re-detect the target object. The resampling stage of the particle filter is modified on the fly according to the occlusion state and different motion models are adopted. To tackle with the partial occlusions, the authors proposed to use a 2D projection confidence map for reducing the contamination from the background. They first used a background detection algorithm [73] to remove the background pixels. Then the foreground pixels in the tracker bounding box is divided into a 3 x 3 grid. For each cell of this grid, the ratio of foreground pixels to the number of all pixels is calculated and the confidence score is obtained by drawing from empirically obtained distributions. The representation of the target is achieved by combining adaptive histogram of colors [91], histogram of depth, template of edges [52], HOG [95], 3D shape parameters [53] and Local Binary Patterns [100]. Despite its impressive accuracy, 0.9 FPS reported by the authors is slow for many real-world applications.

Another particle filter tracker was proposed by Bibi *et al*.in [11]. The authors adopted a 3D part based method and detected the occlusions using the sudden decreases of 3D points in the sampled particles. To obtain a feasible number of particles, they divided each particle cuboid into fixed number of overlapping 3D windows. Then a 13 dimensional feature vector is extracted from each part; 10 color names [127] and 3 3D shape features [53]. Due to the nature of the point clouds, some parts might be empty or near-empty. In order to impose a temporal constraint on the parts, each particle also has an $N-D$ binary vector to encode the emptiness. During particle sampling in the next frame, hamming distance [42] is used for fil-

tering out the unlikely particles. In the absence of occlusion, the authors adopted a 2D optical flow [51] to obtain a rough transformation matrix which is then used for particle sampling. When the target is occluded, a zero-mean motion model is applied. Explicit occlusion detection is achieved by observing the decrease in the 3D points in all particles by using a depth-normalized measure.

Xiao *et al*. [135] proposed a two-layer target representation where a global layer represents the full appearance while the local layer acts as a part based template. In each frame, first the global layer evaluates the candidate regions to find a clear match. In case this condition does not hold, the local layer is used for a part based matching. Two separate KCFs [48] are adopted to apply tracking in the global layer on color and depth images individually where color names [127] and HOG [95] are used as features respectively.

2.4.2 Discriminative Trackers

Camplani *et al*. [86] proposed to use the depth data within a DCF framework in order to achieve fast tracking with explicit occlusion handling. The authors built their tracker upon Kernelized Correlation Filters (KCF) [48] where they used the depth information as an additional target representation, for occlusion detection and for helping accurate scale estimation. To achieve this, a 1-D K-Means clustering similar to [121] is applied to the target region depth histogram which is followed by a connected component analysis to further refine the region separation and outlier exclusion. The occlusion detection is done in a similar spirit with [113] by employing a Gaussian to model the target object's depth distribution and detecting the occluding objects by finding the points that do not fit into this model. However, occlusion recovery significantly differs from [113] by taking advantage of KCF [48]'s speed. Instead of following the occluder with optical flow [20], creating a list of segmented regions and applying SVM [28] to re-detect the target, Camplani *et al*. tracks the occluding object with a separate KCF [48]. A search region is then created using the tracked occluder and depth regions are clustered within. A mean depth value and a tracker response value are calculated from each of these clusters. Tracker resumes tracking the target in case of either the overlap between the best candidate and the tracked occluder drops below a certain threshold or the tracker response on the best candidate is above a threshold. Even though the proposed method achieves real-time

speed, it lags behind significantly in terms of accuracy.

Hannuna *et al*. [43] further extended the work presented in [86] by facilitating a Kalman Filter [56] during occlusion handling and adding a shape analysis stage for better bounding box regression. The authors proposed to add a Kalman Filter to the system and update it during normal tracking by using image plane coordinates and the depth data. This 2.5D information is employed for providing a more precise search space in case of occlusion by using the position and the velocity of the centroids as state variables. Shape analysis on the other hand is done by fitting a bounding box to the depth based foreground segmentation so that the aspect ratio changes can be addressed.

An *et al*. [2] proposed a detection-learning-segmentation framework to tackle the tracking problem in RGBD domain. The authors used KCF [48] as the base tracker for localization and a 1-D Gaussian for modelling the target depth distribution in the segmentation stage. Occlusion is estimated by using the extracted foreground mask by calculating the ratio of the foreground pixels to the total number of pixels in the region of interest.

2.5 Tracking Datasets and Performance Measures

To evaluate different algorithms fairly, publicly available, standard datasets and evaluation protocols have the utmost importance. In this section, we will first introduce the frequently used benchmark datasets and then the evaluation protocols used for measuring the success of the trackers.

2.5.1 RGB Datasets

Earlier attempts for creating tracking datasets focused on surveillance use case due to its real-life applicability. Collins *et al*. [26] compiled a set of sequences along with a GUI based evaluation website. The dataset is comprised of 9 sequences which are mostly captured from the air. Fisher *et al*. [38] created a dataset of 28 sequences varying between 500 - 1400 frames each in a public surveillance setting. Ferryman *et al*. [37] proposed 11 sets where each set consists of with a number of views between 4 - 8. The authors also provided the calibration data between the cameras which fa-

cilitates the usage of this dataset as a multi-view benchmark. Mueller *et al*. [94] proposed a UAV target tracking benchmark with 123 high resolution sequences. These sequences are then annotated with axis-aligned bounding boxes and attributes; *Aspect Ratio Change, Background Clutter, Camera Motion, Fast Motion, Full Occlusion, Illumination Variation, Low Resolution, Out-of-View, Partial Occlusion, Similar Object, Scale Variation, Viewpoint Change*.

Despite these efforts, the aforementioned datasets lack the variety in terms of targets and scenarios. Hence, a need for creating benchmarks that would suit the needs of VOT community arose. In this regard, Wu *et al*. [131, 132] proposed a novel dataset and a unified evaluation methodology specifically suited for generic, single object trackers. The authors collected and axis-aligned annotated 100 targets (the actual number of sequences is smaller than 100 since some sequences have more than one target) in various scenarios. Each sequence is classified with 11 attributes, namely; *Illumination Variation, Scale Variation, Occlusion, Deformation, Motion Blur, Fast Motion, In-Plane Rotation, Out-of-Plane Rotation, Out-of-View, Background Clutter, Low Resolution*.

Smeulders *et al*. [112] created the large scale Amsterdam Library of Ordinary Videos for tracking, ALOV++, which is comprised of 315 sequences mostly crawled from YouTube. A ball, an octopus and microscopic cells are among the 64 different target types. 13 attributes; *Light, Surface Cover, Specularity, Transparency, Shape, Motion Smoothness, Motion Coherence, Clutter, Confusion, Low Contrast, Occlusion, Moving Camera, Zooming Camera* are used to annotate the sequences and the majority of the sequences are relatively short with a mean of 9.2 seconds and maximum 35 seconds. Total number of 89364 frames are annotated on every fifth frame with axis-aligned bounding box unless the motion is extreme. In cases when every fifth frame is insufficient, a denser annotation was preferred.

Recently, Kristan *et al*. [66] started the effort to unify the VOT community by further standardizing the VOT evaluation protocol and organizing annual challenges [61, 62, 63, 65, 87, 88]. The authors proposed a benchmark dataset by gathering commonly used tracking sequences in the community and prepared a compact set of 16 sequences. Each frame is then annotated with axis-aligned bounding boxes and the following visual attributes; *occlusion, illumination change, motion change, size change, camera motion*. If none of these attributes apply to a given frame, it is

labelled as *non-degraded*. Even though the VOT Challenge started as a short-term tracking evaluation, it has evolved to cover different types of tracking over the years such as long-term tracking, real-time tracking, RGBT (thermal imaging) tracking and RGBD tracking [63].

2.5.2 RGBD Datasets

Due to the difficulty of capturing reliable RGBD data, the literature on it is significantly more limited than its RGB counterpart. An early work in tracking on RGBD videos was proposed in [76, 114] by providing only a single sequence with 1132 frames. As the affordable, active RGBD capturing devices have become more accessible, the interest in the field was ignited. First major attempt to create a large scale and diverse benchmark for RGB-D tracking was proposed by Song *et al*. [113]. The authors used a Microsoft Kinect v.1 to capture 100 sequences and manually annotated them with axis-aligned bounding boxes. The videos include a field-of-view between 0.5 to 10 meters with various objects such as humans, toys, animals as the targets. Five attributes were defined as per-sequence attributes; *Target Type: Human, Animal*, *Rigid*, *Target Size: Large, Small*, *Movement: Slow, Fast*, *Occlusion: Yes, No*, *Motion Type: Passive, Active*. Song *et al*. [113] provided the ground truth only for the 5 training sets while sequestering the remaining 95 sets for the evaluation. They created a website [1] where the results can be uploaded to be evaluated by the system and compared against the other entries in the list as a means to prevent overfitting.

Xiao *et al*. [135] proposed a novel RGBD dataset of 36 sequences to further improve the shortcomings of [113] such as compactness and camera motion. The authors used two Asus Xtion devices to capture each scene simultaneously within a range of 0.5 - 8 meters. The mean length of the sequences is 300 frames with 700 frames as maximum. Xiao *et al*. [135] also captured scenes outdoor scenes which was missing in [113] due to the limitations of active RGBD capturing devices. However, since these sequences were also captured with an active sensor, they are rather limited. The sequences then were manually annotated with axis-aligned bounding boxes. Each frame was also labelled with the following attributes; *Illumination Variation, Depth Variation, Scale Variation, Color Distribution Variation, Depth Distribution Variation, Surrounding Depth Clutter, Surrounding Color Clutter, Background*

[1] https://tracking.cs.princeton.edu/index.html

Color Camouflages, Background Shape Camouflages, Partial Occlusion.

2.5.3 Single Object Short-Term Tracking Measures

Since unified benchmarks are a relatively new phenomenon in the VOT community, many different measures have been proposed in the literature. Primitive measures such as Center Error [107] is defined as the $L2$ norm between the ground truth and the tracking result centroids, Tracking Success Probability [70] was proposed by Li *et al.* similar to PASCAL methodology [34]. To provide a better insight for the tracking performance, more complex measures have also been proposed. Nawaz *et al.* [98] fused the tracking failure and tracking accuracy measures into a single number.

Among the more commonly used measures is *Precision Plots* [3]. Precision plots are used to visualize the percentage of frames in which the tracker center error is within a certain threshold. The reason to adopt this measure instead of mean center error is the fact that once a tracker lost the target, the mean center error will be arbitrarily large and will not reflect the true performance. An example can be seen in Figure. 2.4.

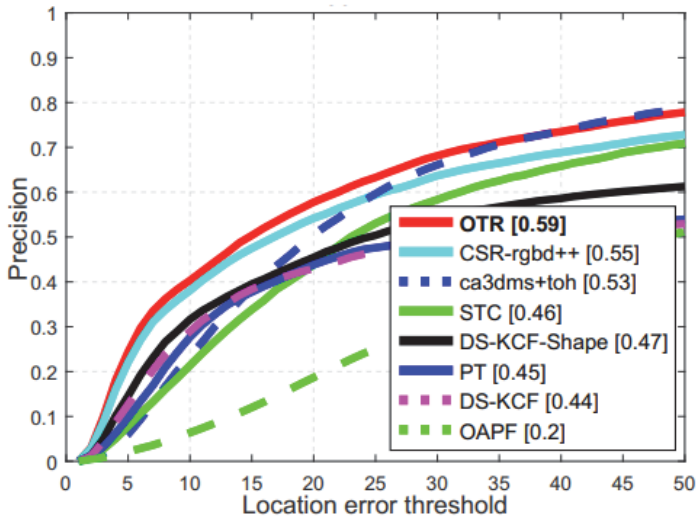

Figure 2.4 An example precision plot [123]

Second common visualization is the *Success Plots* where the average overlap [34] is plotted over a multitude of thresholds to create an Area Under Curve (AUC) as it

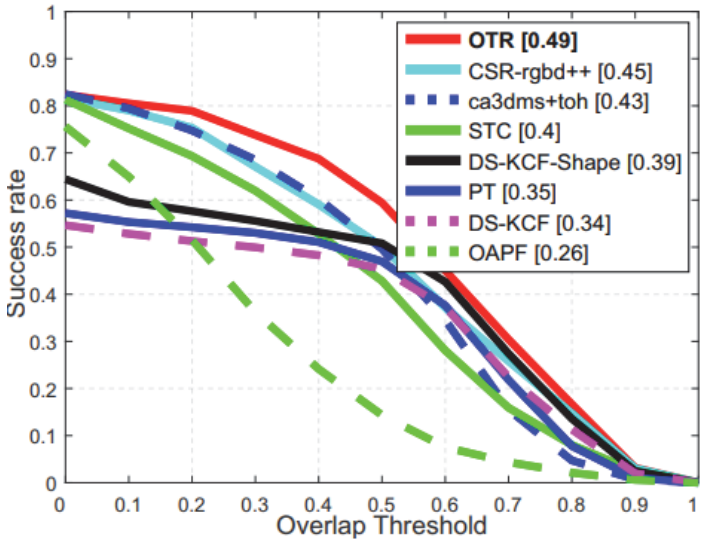

Figure 2.5 An example success plot [123]

is illustrated in Figure. 2.5.

The overlap is defined as

$$S(t) = \frac{r_p^t \cap r_g^t}{r_p^t \cup r_g^t} \tag{2.1}$$

where r_p^t is the region predicted by the tracker and r_g^t is the ground truth region at time t.

An easy way to evaluate tracking algorithms is the One Pass Evaluation (OPE) [131, 132]. This paradigm initializes the tracker on the first frame and lets it run until the end of the sequence. However, to tackle with the possible initialization sensitivity issues, Wu *et al*. also proposed to use two more metrics namely temporal robustness evaluation and spatial robustness evaluation respectively. These metrics aim to perturb the initialization in the spatial and temporal domain to evaluate the trackers more fairly.

Instead of an OPE approach Kristan *et al*. [64] proposed to adopt a multi-pass approach where the tracker is reinitialized once the overlap as given in Eq. 2.1 is zero. This was justified by the fact that the tracker performance can be biased if the target is lost at an earlier stage of the sequence. As the measures, the authors used *accuracy* and *robustness* due to the weak correlation between them. Accuracy is

defined as the average overlap over a sequence and the robustness is defined as how many times the tracker had to be re-initialized.

2.5.4 Single Object Long-Term Tracking Measures

The most commonly used long-term measure for single object trackers was proposed by Lukezic *et al*. [79] and later adopted by the VOT community [62]. The authors proposed to merge three long-term tracking measures into a single number. To this end, they adopted F-Score as the primary measure. Eq. 2.1 is extended to include a classification threshold τ_θ where θ_t is the confidence score from the tracker at time t. In case $\theta_t < \tau_\theta$, the prediction is considered invalid. Hence, Eq. 2.1 is reformulated as $S(r_p^t(\tau_\theta), r_g^t)$.

Precision at a specific threshold τ_θ is defined as

$$Pr(\tau_\theta) = \frac{1}{N_p} \sum_{t \in \{t:\theta_t > \tau_\theta\}} S(r_p^t(\tau_\theta), r_g^t) \tag{2.2}$$

$$Re(\tau_\theta) = \frac{1}{N_g} \sum_{t \in \{t:r_g^t \neq \emptyset\}} S(r_p^t(\tau_\theta), r_g^t) \tag{2.3}$$

where S represents the overlap between two regions, N_g is the number of frames with a ground truth and N_p is the number of predictions.

F-measure is then defined as;

$$F(\tau_\theta) = \frac{2Pr(\tau_\theta)Re(\tau_\theta)}{Pr(\tau_\theta) + Re(\tau_\theta)} \tag{2.4}$$

2.5.5 Multi-Object Tracking Measures

The standard evaluation metrics for multiple object trackers are defined within the classification of events, activities, and relationships protocol [8]. The first metric defined in this framework are multiple object tracking precision (MOTP)

$$MOTP = \frac{\sum_{i,t} d_i^t}{\sum_t c^t} \tag{2.5}$$

which calculates the total error for the predicted positions averaged by the total

number of matches. c^t is the number of matches at time t, d_i^t is the distance between the object and its hypothesis and g^t is the total number of objects present in all frames.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{2.6}$$

with m_t is the number of misses, fp_t is the false positives and mme_t is the mismatches.

3 PRELIMINARIES

This section focuses on the mathematical fundamentals used in DCF based tracking by providing an in-depth historical development of it. We first explain the most primitive DCF tracking approach proposed by Bolme *et al*. [13] then proceed to the mathematical connection between DCF tracking and ridge regression proven by Henriques *et al*. [48] followed by spatially constrained DCFs proposed by Galoogahi *et al*. [40]. Finally, we explain the building blocks of the baseline single object tracking paradigm used in this thesis proposed by Lukezic *et al*. [80].

For the sake of readability, we define our own notation for the commonly used variables in Table 3.1 to express the papers uniformly. In addition to the variables defined in Table 3.1, $\odot$ symbolizes term-by-term product, $*$ stands for complex conjugate operation and M is the binary mask used in spatially constrained correlation filters.

Table 3.1 Uniform notation set for the commonly used variables.

Variable	Spatial Domain	Fourier Domain
Features	x	X
Regression Targets / Correlation Results	y	Y
Correlation Filter	h	H

3.1 The Original Discriminative Correlation Filter Based Tracking

DCF based tracking localizes the target by correlating a trained filter with an image patch in which the target is sought. This is achieved by taking the spatial location

with the maximum correlation response score. Due to the duality of correlation as term-by-term multiplication in Fourier domain, tracking is often done with Fourier transformations of the filter and image patch for computational efficiency. Hence, it is defined as;

$$Y = X \odot H^* \tag{3.1}$$

In its most basic form [13], assume that we are given a sequence S with an axis-aligned bounding box as the target location BB_{target} on the initial frame at $t = 0$. A DCF can be trained over samples X_i by minimizing the following equation;

$$\min_{H^*} \sum_i |X_i \odot H^* - Y_i|^2 \tag{3.2}$$

Training set X_i are obtained by cropping a region around BB_{target} and applying small affine transformations. Y_i is the set of desired ground truth which are generally 2D Gaussians with small variance (*e.g.* $\sigma = 2.0$) and their peaks correspond to the target's spatial location.

Solving Eq. 3.2 can be achieved via the closed-form solution;

$$H^* = \frac{\sum_i Y_i \odot X_i^*}{\sum_i X_i \odot X_i^*} \tag{3.3}$$

On the following frames, the localization is then achieved by applying Eq. 3.1 and selecting the point with the maximum score.

Since the target undergoes appearance changes due to illumination, rotation etc. during tracking, it is important to update the tracker to maintain an up-to-date model of the target. This is done by adopting a moving average approach to combine past and present appearances. Consequently on every frame, the following is calculated similar to Eq. 3.3;

$$H_t^* = \frac{A_t}{B_t} \tag{3.4}$$

$$A_t = \eta Y_t \odot X_t^* + (1-\eta) A_{t-1} \tag{3.5}$$

$$B_t = \eta X_t \odot X_t^* + (1-\eta) B_{t-1} \tag{3.6}$$

where η is the learning rate.

Please note that as pre-processing, log transformation is applied to the pixel-values and followed by a normalization between 0.0 and 1.0 for low-contrast scenarios. Additionally, the normalized values are multiplied with a cosine window to reduce the boundary effects caused by the nature of DFT as proposed by Bolme *et al.* [13]. Boundary effects happen since DFT by definition works on periodic signals and applies circular convolution. Therefore, the signal is convolved in a toroid-like structure which connects edges of the image in 2D case. Thus, to avoid discontinuity at the edges, applying a cosine window effectively assigns zero to boundary pixels which facilitates a smooth transition between two periods.

3.2 The Relationship Between Ridge Regression and Discriminative Correlation Filter Tracking

Following Bolme *et al.* [13], Henriques *et al.* [48] showed that DCF tracking is a special case of ridge regression. Assume that we have a ridge regression problem defined as the following;

$$\min_{w} \sum_{i} (f(x_i) - y_i)^2 + \lambda ||w||^2 \tag{3.7}$$

where x_i are the training samples, y_i are the regression targets, $f(z) = w^T z$ and λ is the regularization parameter. The closed-form solution to Eq. 3.7 is provided by Rifkin *et al.* [68] as;

$$w = (\hat{x}^T \hat{x} + \lambda I)^{-1} \hat{x}^T \hat{y} \tag{3.8}$$

with x_i constitutes the rows of $\hat{x}$, y_i constitutes the rows of $\hat{y}$ and I is the identity matrix. In Fourier domain, this equation becomes

$$w = (\hat{X}^H \hat{X} + \lambda I)^{-1} \hat{X}^H \hat{Y} \tag{3.9}$$

with $\hat{X}^H = (\hat{X}^*)^T$.

Henriques *et al.* [48] points to the fact that if $\hat{x}$ is a circulant matrix, *e.g.* x_i are spatially shifted versions of a base sample x, then applying DFT converts X into a

diagonal matrix.

$$X' = F\,\mathrm{diag}(X)F^H \tag{3.10}$$

where F is DFT matrix. Then $X'^H X'$ becomes

$$X'^H X' = F\,\mathrm{diag}(X^*)F^H F\,\mathrm{diag}(X)F^H \tag{3.11}$$

which can be further simplified to

$$X'^H X' = F\,\mathrm{diag}(X^*)\mathrm{diag}(X)F^H \tag{3.12}$$

Since $F^H F = I$ and element-wise products can be used on diagonal matrices, Eq. 3.12 can be rewritten as

$$X'^H X' = F\,\mathrm{diag}(X^* \odot X)F^H \tag{3.13}$$

where $X^* \odot X$ is the auto-correlation of x.

Putting Eq. 3.13 into Eq. 3.9,

$$W = \frac{X^* \odot Y}{X^* \odot X + \lambda} \tag{3.14}$$

is obtained.

One can immediately see the similarity between Eq. 3.3 and Eq. 3.14 and the main difference here is that Eq. 3.3 is solved over multiple training samples whereas Eq. 3.14 is solved over a single sample. However, more importantly, Henriques *et al*. [48] proves mathematically that DCF tracking in Fourier domain is actually ridge regression whereas Bolme *et al*. [13] provides a formula in Fourier domain in an ad-hoc way. Please note that spatially shifted training samples x_i are implicitly provided by the nature of circular correlation/convolution operation.

3.3 Spatially Constrained Discriminative Correlation Filters

Despite the mathematical elegance of Henriques *et al*. [48]'s proposal, Eq. 3.7 suffers from so-called *boundary effects*. Boundary effects defines the phenomenon of spatial circularity of the training samples during DCF training which results with

unrealistic negative samples as it can be seen in Figure. 2.2.

To show this more clearly, Galoogahi *et al*. [40] first explicitly expressed the ridge regression problem in the spatial domain as following;

$$E(h) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D}||y_i(j) - h^T x_i[\Delta\tau_j]||_2^2 + \frac{\lambda}{2}||h||_2^2 \tag{3.15}$$

where $y_i \in \mathbb{R}^D$ are the regression targets for the i-th sample, $x_i \in \mathbb{R}^D$ is the i-th sample and λ is the regularization parameter while $\mathbb{C} = [\Delta\tau_1, ..., \Delta\tau_D]$ stands for the cyclic shifts of the input signal. Solution for Eq. 3.15 then becomes

$$h = H^{-1}\sum_{i=1}^{N}\sum_{j=1}^{D} y_i(j)x_i[\Delta\tau_j] \tag{3.16}$$

with

$$H = \lambda I + \sum_{i=1}^{N}\sum_{j=1}^{D} x_i[\Delta\tau_j]x_i[\Delta\tau_j]^T \tag{3.17}$$

which has a complexity of $\mathcal{O}(D^3 + ND^2)$.

On the other hand, in order to avoid the boundary effects, a training signal $x \in \mathbb{R}^T$ larger than the filter $h \in \mathbb{R}^D$ can be adopted via using a binary masking matrix M where $T > D$.

Then Eq. 3.15 can be written as

$$E(h) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{T}||y_i(j) - h^T M x_i[\Delta\tau_j]||_2^2 + \frac{\lambda}{2}||h||_2^2 \tag{3.18}$$

In case when $T >> D$, the boundary effects can be avoided for the most of the training samples at a computational cost of $\mathcal{O}(D^3 + NTD)$.

Solving Eq. 3.15 in Fourier Domain produces the same result as Eq. 3.14 with the complexity of $\mathcal{O}(NDlogD)$. Applying Fourier Transformation to Eq. 3.18 results with

$$E(h) = \frac{1}{2}\sum_{i=1}^{N}||Y_i - \text{diag}(X_i)^T\sqrt{D}FM^T h||_2^2 + \frac{\lambda}{2}||h||_2^2 \tag{3.19}$$

However, because of the spatial constraint, h must be solved in spatial domain

with a complexity of $\mathcal{O}(D^3 + ND^2)$.

To solve Eq. 3.19 efficiently, Galoogahi *et al.* [40] proposed to adopt the Augmented Lagrangian Method [16]. With the help of an auxiliary variable G, Eq. 3.19 is reformulated as

$$\begin{aligned} E(h,G) &= \frac{1}{2}\sum_{i=1}^{N}||Y_i - \text{diag}(X_i)^T G||_2^2 + \frac{\lambda}{2}||h||_2^2 \\ G &= \sqrt{D}FM^T h \end{aligned} \tag{3.20}$$

Lagrangian then becomes;

$$\begin{aligned} L(G,h,Z) = &\frac{1}{2}\sum_{i=1}^{N}||Y_i - \text{diag}(X_i)^T G|| + \frac{\lambda}{2}||h||_2^2 \\ &+ Z^T(G - \sqrt{D}FM^T h) \\ &+ \frac{\mu}{2}||G - \sqrt{D}FM^T h||_2^2 \end{aligned} \tag{3.21}$$

with μ is the penalty constant and Z is Fourier transform of the Lagrangian vector.

Subproblem g can be solved as following

$$\begin{aligned} G^* &= \arg\min L(G;H,Z) \\ &= (S_{xy} + \mu H - Z) \odot^{-1} (S_{xx} + \mu \mathbb{1}) \end{aligned} \tag{3.22}$$

where S_{xx} and S_{xy} are defined as

$$S_{xx} = \sum_{i=1}^{N} X_i \odot X_i^* \quad \& \quad S_{xy} = \sum_{i=1}^{N} Y_i \odot X_i^* \tag{3.23}$$

$H = \sqrt{D}FM^T h$ and $\odot^{-1}$ is element-wise division. However, H is estimated by FFT of h and padding from M^T.

Subproblem h on the other hand can be defined as

$$\begin{aligned} h^* &= \arg\min L(h; g, l) \\ &= (\mu + \frac{\lambda}{\sqrt{D}})^{-1}(\mu g + l) \end{aligned} \tag{3.24}$$

with $g = \frac{1}{\sqrt{D}} MF^T G$ and $l = \frac{1}{\sqrt{D}} MF^T Z$. Similar to g, h is estimated by an inverse FFT and applying the lookup table from M.

Lagrangian multiplier update is then defined as

$$Z^{(i+1)} \leftarrow Z^{(i)} + \mu(G^{(i+1)} - H^{(i+1)}) \tag{3.25}$$

where i is the $i-th$ iteration.

Finally μ is calculated by

$$\mu^{(i+1)} = \min(\mu_{\max}, \beta\mu^{(i)}) \tag{3.26}$$

with β is a constant.

Putting all together, pseudocode for finding h using Alternating Direction Method of Multipliers (ADMM) is given below;

Algorithm 1 Calculating h using ADMMs [40]

Initialize $h^{(0)}$, $l^{(0)}$
Pad with zeros and apply FFT: $\sqrt{D}FM^{\mathrm{T}}h^{(0)} \rightarrow H^{(0)}$
Apply FFT: $\sqrt{D}Fl^{(0)} \rightarrow Z^{(0)}$
Calculate S_{xx} and S_{xy} via Eq. 3.23
$i = 0$
while G, h, Z **has not converged**
  Calculate $G^{(i+1)}$ via Eq. 3.22, $H^{(i)}$ & $Z^{(i)}$
  IFFT and crop $\frac{1}{\sqrt{D}} MF^{\mathrm{T}}G^{(i+1)} \rightarrow g^{(i+1)}$
  IFFT and crop $\frac{1}{\sqrt{D}} MF^{\mathrm{T}}Z^{(i+1)} \rightarrow l^{(i+1)}$
  Calculate $h^{(i+1)}$ via Eq. 3.24 $g^{(i+1)}$ & $l^{(i)}$
  Pad and apply FFT: $\sqrt{D}FM^{\mathrm{T}}h^{(i+1)} \rightarrow H^{(i+1)}$
  Update Lagrange vector via Eq.3.25
  Update μ via Eq. 3.26
  $i = i + 1$
end

3.4 Spatially Constrained Discriminative Correlation Filters With Reliability Maps

Building on top of Galoogahi *et al*. [40]'s idea of using spatial constraints, Lukezic *et al*. [80] proposed to adopt color cues in the images to obtain a probabilistic spatial reliability map instead of a fixed, rectangular region. To this end, they swapped the binary mask M in Eq. 3.18 with an adaptive, appearance model based reliability map.

Assume a per-pixel reliability map $M \in [0,1]^{WxH}$ with each element $M \in \{0,1\}$, then reliability is calculated via

$$p(M=1|y,x) \propto p(y|M=1,x)p(x|M=1)p(M=1) \tag{3.27}$$

where y is the appearance of pixel x and $p(y|M=1,x)$ is calculated via Bayes rule based on the foreground/background histograms that are extracted during the tracking process. $p(x|M=1)$ is the probability of a pixel belonging to the foreground (mask value is 1) and $p(M=1)$ is the ratio of foreground/background region size at the time of histogram extraction.

As a robustness measure against the deformations, rotations etc. the authors adds more emphasis to the central elements with a prior

$$p(x|M=1) = k(x;\sigma) \tag{3.28}$$

with $k(x;\sigma)$ a modified Epanechnikov kernel defined as

$$k(x;\sigma) = 1-(r/\sigma)^2 \tag{3.29}$$

where σ is a tunable parameter.

As a further improvement to Galoogahi *et al*. [40], Lukezic *et al*. [80] also proposes to take advantage of response scores from individual feature channels as channel reliability scores at learning and detection stages. The calculation of individual

channels' reliability scores for learning stage then becomes;

$$w_d = Z \max(f_d * h_d) \tag{3.30}$$

where d symbolizes the $d-th$ feature channel, $\mathbf{f}$ stands for the features, $\mathbf{h}$ is the correlation filter, $*$ symbolizes correlation operation and Z is the normalization factor with

$$\sum_d w_d = 1 \tag{3.31}$$

Detection stage channel reliability score on the other hand is calculated similar to Bolme *et al*. [13]'s Peak to Side Lobe Ratio (PSR) based approach. Assume that ρ_{max1} and ρ_{max2} are the highest two peaks in the response map obtained as a result of $\mathbf{f_d} * \mathbf{h_d}$. Channel reliability score in detection mode is calculated as

$$w_d^{(det)} = 1 - \arg\min(\rho_{max2}/\rho_{max1}, \frac{1}{2}) \tag{3.32}$$

Pseudocode for the tracking process is then

Algorithm 2 Tracking with Spatial and Channel Reliability

Require:

$\mathbf{I_t}$: Image at time t, $\mathbf{p_{t-1}}$: target position at $t-1$, $\mathbf{s_{t-1}}$: scale at $t-1$, $\mathbf{h_{t-1}}$: filter at $t-1$, $\mathbf{c_{t-1}}$: color histograms at $t-1$, $\mathbf{w_{t-1}}$: channel reliability at $t-1$

Localization and Scale Estimation:

$\mathbf{p_t}$: estimated using $\mathbf{h_{t-1}}$ and $\mathbf{f_t}$: features at t and $\mathbf{w_{t-1}}$

$\mathbf{s_t}$: scale at t is estimated using the $\mathbf{p_t}$

Update:

Extract foreground and background histograms $\tilde{\mathbf{c}}^{\mathrm{f}}$, $\tilde{\mathbf{c}}^{\mathrm{b}}$

Update the histograms

$\mathbf{c_t}^{\mathrm{f}} = (1-\eta_c)\mathbf{c}^{\mathrm{f}}{}_{\mathbf{t-1}} + \eta_c \tilde{\mathbf{c}}^{\mathrm{f}}$

$\mathbf{c_t}^{\mathrm{b}} = (1-\eta_c)\mathbf{c}^{\mathrm{b}}{}_{\mathbf{t-1}} + \eta_c \tilde{\mathbf{c}}^{\mathrm{b}}$

Calculate the spatial reliability map M_t at t

Calculate the new filter $\tilde{\mathbf{h}}$ using M_t

Calculate the new channel reliability $\tilde{\mathbf{w}}$ using $\tilde{\mathbf{h}}$

Update the filter $\mathbf{h_t} = (1-\eta)\mathbf{h_{t-1}} + \eta\tilde{\mathbf{h}}$

Update the channel reliability $\mathbf{w_t} = (1-\eta)\mathbf{w_{t-1}} + \eta\tilde{\mathbf{w}}$

4 CONTRIBUTIONS

This thesis proposes three novel long-term, discriminative, RGBD, single object tracker methods to include depth information in DCF tracking and one novel RGBD dataset. As a compendium type of doctoral thesis, this chapter will summarize the main findings of the publications enclosed with the thesis while the details can be found in the publications. The publications are explained in chronological order and will be referred as [P1], [P2], [P3] and [P4].

4.1 P1 Depth Masked Discriminative Correlation Filter

In this paper, spatially constrained DCF with reliability maps that was explained in Chapter 3.3 are extended into RGBD domain. This is accomplished by leveraging the depth data which provides accurate information about the object's 3D position [58].

The contributions are twofold;

- Spatial constraints using depth masking
- A novel occlusion handling method based on object's distribution and DCF response history to avoid target model pollution

The first contribution is achieved by replacing probabilistic distribution for spatial support defined in Eq. 3.27 by the depth value distributions of foreground and background pixels. The distributions are modelled using separate single Gaussians $P_{fg} \propto \mathcal{N}(\mu_{fg}, \sigma^2_{fg})$ and $P_{bg} \propto \mathcal{N}(\mu_{bg}, \sigma^2_{bg})$ which are updated per-frame basis to maintain an up-to-date models using the following;

$$\begin{aligned} \mu^{(t)} &= \mu^{(t)}\theta + (\mu^{(t-1)}(1-\theta)) \\ \sigma^{(t)} &= \sigma^{(t)}\gamma + (\sigma^{(t-1)}(1-\gamma)) \end{aligned} \tag{4.1}$$

where θ and γ are fixed hyperparameters. This formulation of DCF tracking provides an inherent adoption of depth data in spatially constrained DCF framework.

As the second contribution, occlusion detection is implemented as a combination of two weak classifiers; tracker response values and depth based segmentation.

Basis of tracker response value approach is defined as;

$$r_{max}^{(t+1)} = r_{max}^{(t)} + \frac{r_{max}^{curr} - r_{max}^{t}}{t} \tag{4.2}$$

where r_{max}^{curr} is the maximum response value of the tracker output on the latest frame. This approach is intuitive however, it also risks false negatives/positives if the tracker model has already been corrupted before an occlusion event.

To further strengthen the failure detection mechanism, a depth segmentation trigger is also added. In the ideal case, all pixels in the tracker bounding box should belong to the target (foreground). Leveraging this information, a 10% threshold as the minimum viable ratio of foreground pixels to the bounding box area is defined.

The overall structure of the proposed algorithm is given in Figure. 4.1.

Pseudocode for the proposed tracker given in Algorithm 3.

The proposed algorithm has been evaluated on the Princeton RGBD benchmark [113] using an online evaluation protocol which is based on PASCAL VOC measure [34] provided by Song *et al.* [113] and the results on the day of paper submission are given as follow in Table 4.1;

Table 4.1 Experiments on the Princeton Tracking Benchmark using the PTB protocol. Numbers in the parenthesis are the ranks[P1].

	Avg Rank	Tracking Category											
Method		*Human*	*Animal*	*Rigid*	*Large*	*Small*	*Slow*	*Fast*	*Occ.*	*No-Occ.*	*Passive*	*Active*	*FPS*
3D-T [11]	2.81	**0.81 (1)**	0.64 (4)	0.73 (5)	**0.80 (1)**	0.71 (3)	0.75 (5)	**0.75 (1)**	**0.73 (1)**	0.78 (5)	0.79 (3)	0.73 (2)	N.A
RGBDOcc+OF [113]	3.27	0.74 (4)	0.63 (5)	**0.78 (1)**	0.78(3)	0.70 (4)	0.76 (2)	0.72 (3)	0.72 (2)	0.75 (6)	0.82 (2)	0.70 (4)	0.26
OAPF [92]	3.45	0.64 (6)	**0.85 (1)**	0.77 (3)	0.73 (5)	0.73 (2)	**0.85 (1)**	0.68 (6)	0.64 (6)	**0.85 (1)**	0.78 (4)	0.71 (3)	0.9
Our	3.63	0.76 (3)	0.58 (6)	0.77 (2)	0.72 (6)	**0.73 (1)**	0.75 (4)	0.72 (4)	0.69 (3)	0.78 (4)	**0.82 (1)**	0.69 (6)	**8.3**
DLST [2]	3.63	0.77 (2)	0.69 (3)	0.73 (6)	0.80 (2)	0.70 (6)	0.73 (6)	0.74 (2)	0.66 (4)	0.85 (2)	0.72 (6)	**0.75 (1)**	4.6
DS-KCF-Shape [43]	4.18	0.71 (5)	0.71 (2)	0.74 (4)	0.74 (4)	0.70 (5)	0.76 (3)	0.70 (5)	0.65 (5)	0.81 (3)	0.77 (5)	0.70 (5)	35.4
CSR-DCF [80]	10.55	0.53 (9)	0.56 (11)	0.68 (12)	0.55 (12)	0.62 (9)	0.66 (12)	0.56 (10)	0.45 (14)	0.79 (6)	0.67 (12)	0.56 (9)	13.6

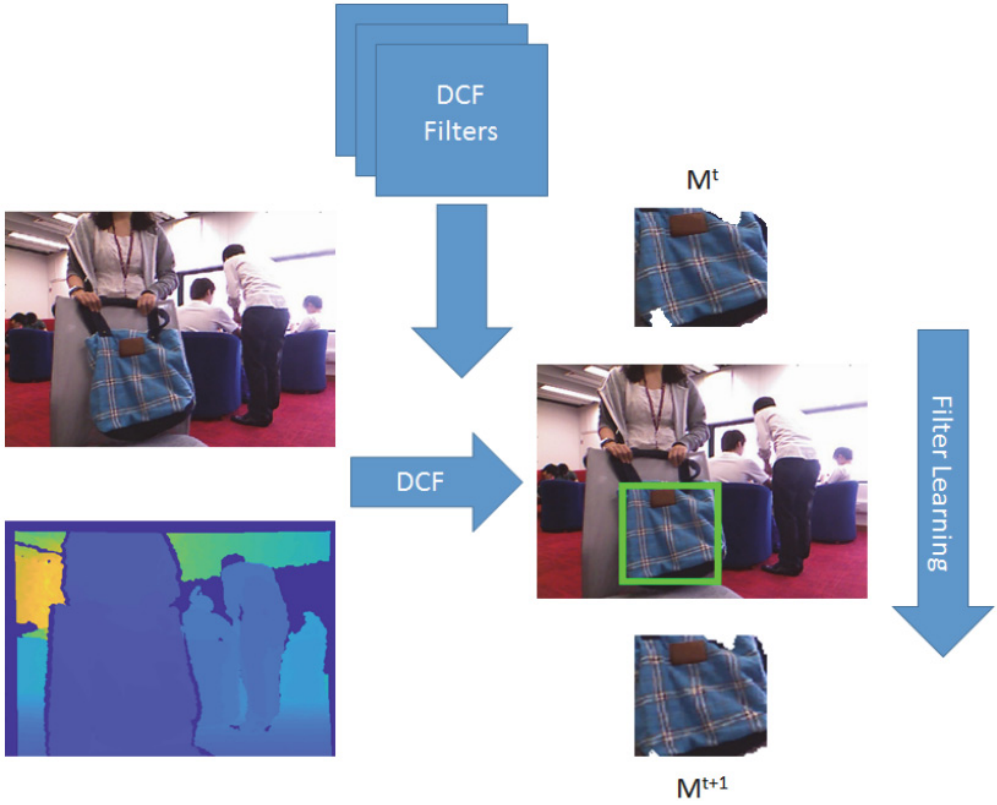

Figure 4.1 The overview for the proposed algorithm in [P1].

As it can be seen from the Table 4.1, the depth information drastically improves the results compared to the baseline tracker CSR-DCF [80] while providing a decent frame rate. This proves that adopting depth information in spatially constrained DCF context has significant potential for RGBD tracking since it provides an additional discriminative layer to the tracking framework. A visual example for the benefits of depth is given in Figure. 4.2.

Algorithm 3 Depth Masked DCF [58]

Require: Current frame I^t; Occlusion state S^{t-1}; Foreground and background depth distributions P_{fg}^{t-1}, P_{bg}^{t-1}; Tracker response threshold τ; K last responses $\vec{r}$

if S^{t-1} is $false$ **then** {** Tracker part **}
  Run DCF tracker ($\mathbf{h^{t-1}}$) on I^t
  Calculate maximum filter response r_{max}^t
  Run occlusion detection to obtain S^t
  Calculate depth mask $\mathbf{M^t}$
else {** Detector part **}
  Run full frame detection and obtain r_{max}^t
  if $r_{max}^t > \tau * mean(\vec{r})$ **then**
    $S^t \leftarrow false$
  else
    $S^t \leftarrow true$
  end if
end if
if S^t is $false$ **then** {** Mask update part **}
  Update distributions P_{fg}^t and P_{bg}^t using $\mathbf{M^t}$
  Update $\mathbf{h^t}$ using $\mathbf{M^t}$
  Update tracker response history $\vec{r}^{\,mod(t,K)} \leftarrow r_{max}^t$
end if
Proceed to the next frame

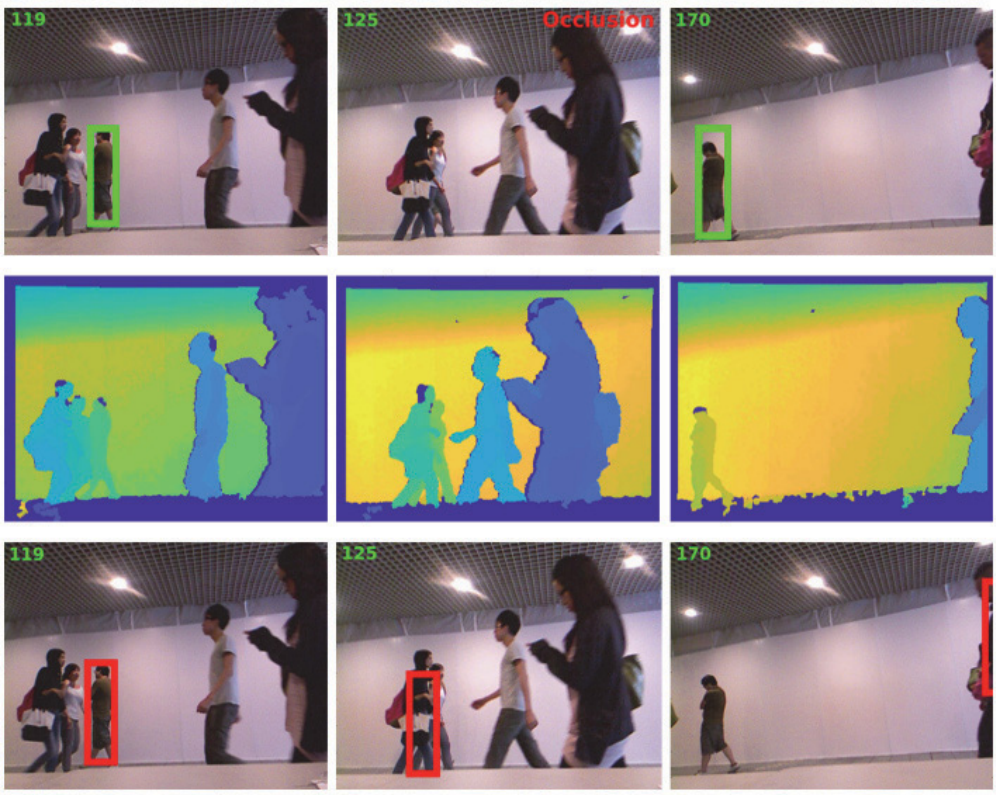

Figure 4.2 The top row is the tracking results for the proposed algorithm whereas the bottom row is for the baseline tracker that works on RGB channels. The baseline algorithm drifts in the presence of an occluder while the proposed algorithm is able to detect the occlusion and starts the re-detection module using the depth information and localizes the target once it becomes visible [P1].

4.2 P2 How to Make an RGBD Tracker ?

In this paper, a novel, generic framework to convert any short-term RGB tracker into an RGBD tracker is proposed [57]. In order for an RGB tracker to be minimally compatible with this framework, two mild assumptions are done;

- Short-term tracker provides a bounding box
- Tracker's model update can be stopped and restarted

Additionally, if the baseline tracker accepts an external mask for the foreground region similar to the spatially constrained DCFs, the second-level (full) integration can be achieved.

The first level of integration works as a tracking supervisor via a foreground segmentation approach. Similar to [P1], the tracker model update is stopped if the ratio of foreground pixels to the tracker bounding box region drops below a certain threshold and the tracker goes into recovery mode to search for the target. This search is done using three principles; (i) target object must be close to the latest known location, (ii) the latest reliable tracker response should not differ significantly from the recovered object's, (iii) the search region is expanded proportionally to the object's latest known speed. This adaptive approach is more efficient and drastically different to the one adopted in [P1] since [P1] used a naive, full-frame search.

The foreground segmentation is achieved by building on top of [P1]; the proposed framework takes advantage of multiple information sources instead of only depth to obtain an accurate segmentation for the foreground object. To efficiently compute this, the energy minimization formula proposed in [17] is used;

$$E(f) = E_{smooth}(f) + E_{data}(f) \tag{4.3}$$

where the aim is to label each pixel by minimizing the energy. E_{smooth} is the smoothness prior which assigns a high energy if neighboring pixels have different labels [33], E_{data} is based on RGBD and formulated as;

$$E_{data}(f) = E_{color}(f) + E_{depth}(f) + E_{spatial}(f) \tag{4.4}$$

The overview of the proposed algorithm is illustrated in Figure. 4.3.

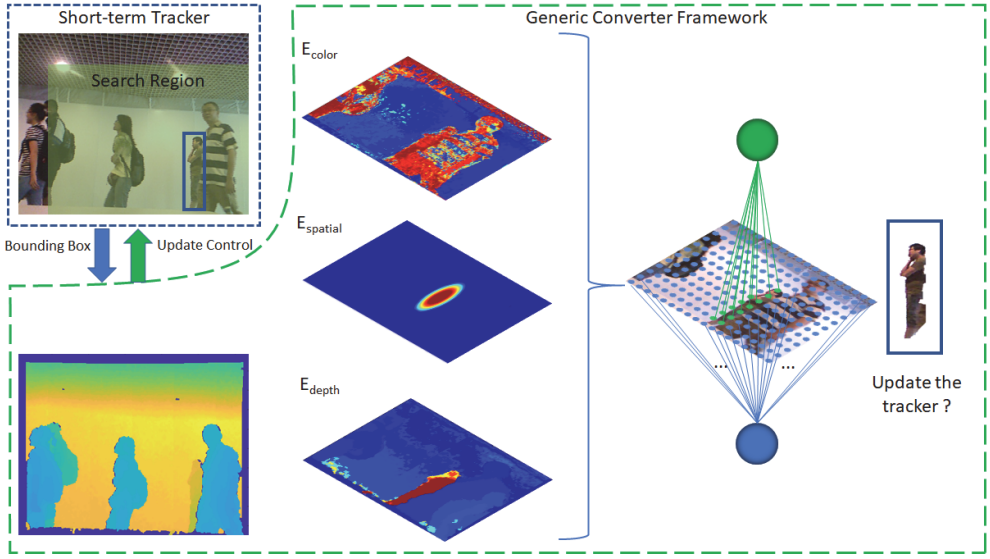

Figure 4.3 The overview for the proposed algorithm in [P2].

To evaluate the proposed framework's effectiveness, experiments on Princeton RGBD Dataset [113] are conducted and the results can be found in Table 4.2.

Table 4.2 Comparison of short-term RGB and RGBD tracking methods on the Princeton Tracking Benchmark (PTB) [113]. DCF [40] and three state-of-the-art trackers were used within the framework – ECOgpu [31], ECOhc [31] and CSR-DCF [80]; their level-one RGBD extensions are denoted DCF-rgbd, ECO-rgbd and CSR-DCF-rgbd, the level-two CSR-DCF integration where the original RGB-based mask is replaced by the proposed foreground mask is denoted CSR-DCF-rgbd++. (The table shows results for the Princeton Benchmark as of June 15, 2018) [P2]

	Avg Rank	Tracking Category										
Method		*Human*	*Animal*	*Rigid*	*Large*	*Small*	*Slow*	*Fast*	*Occ.*	*No-Occ.*	*Passive*	*Active*
CSR-DCF-rgbd++	**3.64**	0.77(2)	0.65(5)	0.76(6)	0.75(4)	**0.73(1)**	0.80(3)	0.72(3)	0.70(3)	0.79(5)	0.79(5)	0.72(3)
OAPF [92]	5.27	0.64(14)	**0.85(1)**	0.77(4)	0.73(6)	0.73(2)	**0.85(1)**	0.68(8)	0.64(8)	**0.85(1)**	0.78(9)	0.71(4)
3D-T [11]	5.36	**0.81(1)**	0.64(7)	0.73(15)	**0.80(1)**	0.71(6)	0.75(8)	**0.75(1)**	**0.73(1)**	0.78(11)	0.79(6)	0.73(2)
RGBDOcc+OF [113]	5.55	0.74(5)	0.63(9)	0.78(2)	0.78(3)	0.70(7)	0.76(5)	0.72(4)	0.72(2)	0.75(17)	0.82(2)	0.70(5)
ECOhc-rgbd	6.18	0.70(7)	0.55(15)	**0.81(1)**	0.69(9)	0.72(4)	0.78(4)	0.68(7)	0.65(6)	0.79(6)	**0.83(1)**	0.66(8)
DSKCF-Shape [43]	6.64	0.71(6)	0.71(3)	0.74(11)	0.74(5)	0.70(8)	0.76(6)	0.70(6)	0.65(7)	0.81(4)	0.77(11)	0.70(6)
DLST [2]	6.73	0.77(3)	0.69(4)	0.73(16)	0.80(2)	0.70(9)	0.73(14)	0.74(2)	0.66(5)	0.85(2)	0.72(16)	**0.75(1)**
DM-DCF [58]	6.73	0.76(4)	0.58(12)	0.77(5)	0.72(8)	0.73(3)	0.75(10)	0.72(5)	0.69(4)	0.78(13)	0.82(3)	0.69(7)
DSKCF [86]	9.36	0.67(10)	0.61(10)	0.76(8)	0.69(10)	0.70(10)	0.75(9)	0.67(9)	0.63(9)	0.78(12)	0.79(7)	0.66(9)
ECOgpu-rgbd	9.82	0.66(11)	0.58(11)	0.76(7)	0.65(14)	0.71(5)	0.81(2)	0.64(14)	0.62(10)	0.77(14)	0.78(8)	0.65(12)
DSKCF-CPP [86]	10.36	0.65(12)	0.64(8)	0.74(12)	0.66(13)	0.69(12)	0.76(7)	0.65(13)	0.60(12)	0.79(7)	0.80(4)	0.64(14)
RGBD+OF [113]	11.36	0.64(15)	0.65(6)	0.75(9)	0.72(7)	0.65(17)	0.73(15)	0.66(10)	0.60(13)	0.79(8)	0.74(15)	0.66(10)
hiob [115]	11.64	0.53(19)	0.72(2)	0.78(3)	0.61(16)	0.70(11)	0.72(16)	0.64(15)	0.53(16)	0.85(3)	0.77(12)	0.62(15)
CSR-DCF-rgbd	11.91	0.68(9)	0.57(13)	0.74(10)	0.68(11)	0.68(14)	0.74(12)	0.65(12)	0.62(11)	0.75(16)	0.77(10)	0.64(13)
ECOhc [31]	12.18	0.69(8)	0.56(14)	0.72(17)	0.67(12)	0.68(13)	0.74(11)	0.65(11)	0.59(14)	0.78(9)	0.74(14)	0.65(11)
ECOgpu [31]	15.36	0.58(16)	0.54(16)	0.73(13)	0.59(18)	0.65(15)	0.73(13)	0.58(17)	0.51(17)	0.78(10)	0.69(17)	0.60(17)
DCF-rgbd	15.45	0.64(13)	0.54(17)	0.73(14)	0.65(15)	0.65(16)	0.71(17)	0.63(16)	0.59(15)	0.74(18)	0.76(13)	0.61(16)
DCF [40]	18.09	0.56(17)	0.52(19)	0.66(18)	0.60(17)	0.59(19)	0.65(18)	0.57(18)	0.48(18)	0.74(19)	0.68(18)	0.56(18)
CSR-DCF [80]	18.36	0.54(18)	0.53(18)	0.64(19)	0.56(19)	0.59(18)	0.61(19)	0.56(19)	0.44(19)	0.76(15)	0.64(19)	0.55(19)

Table 4.2 clearly shows that adding depth data to the state-of-the-art RGB trackers (level one integration) provides a significant performance boost in terms of accuracy. Moreover, adopting the second level integration when possible increases the performance further and outperforms state-of-the-art trackers.

4.3 P3 Object Tracking by Reconstruction with View-Specific Discriminative Correlation Filters

Despite the state-of-the results, the tracker proposed in [P2] still treated the 3D target objects on a 2D plane. This approach has major limitations and cannot cope well with the out-of-plane rotations.

Consider a book as the target object and assume that we initialize the tracker with its front cover. Once the book starts rotating around itself, gradually the initially seen view (the front cover) will be replaced by its narrow sideways and we will end up with the back cover being visible. Continuous model updating mechanisms proposed in modern trackers including [P2] will first replace the previously seen view (the front cover) with the narrow sideways and finally with the back cover. Thus, the tracker currently only recognizes the back cover as the target object. Assume that after this rotation, the target is occluded and the tracker goes into the occlusion state in which it continuously tries to re-locate the target in a similar manner to what is proposed in [P2]. In case the target has rotated during occlusion and now only the front cover is visible, it is highly possible that the tracker will not be able to re-detect the target since from the tracker's point of view, the appeared object is a completely different object than what it has previously seen.

Detecting this change in 2D is inherently difficult since it is hard to distinguish the appearance changes (blur, illumination etc.) and occlusions from out-of-plane rotation. However, with the help of 3D information, this can be achieved with relative ease. In this paper [123]([P3]), a 3D object representation is employed to address these issues by modelling the target object as a set of *surfels*; 3D points with color, radius and the normal. Surfel based approaches have already been successfully adopted in Simultaneous Localization and Mapping (SLAM) community [109] in which Iterative Closest Point (ICP) is used for updating the 3D structure.

The main contribution in this paper is the idea of using 3D object models in conjunction with a spatially constrained DCF formulation. The benefits to this approach are two-fold;

- 2D projections from 3D models provide better target region representation for the tracker updates
- Detecting out-of-plane rotations using these projections facilitates a *multi-view*

DCF framework which helps more accurate tracking and re-detection capability

To achieve these, a two-level abstraction for the target model is adopted. First level consists of a set of view-specific DCFs, $\{\mathbf{h^s}\}_{s=1}^{S}$, that are stored throughout the tracker's life cycle and each one of them encodes a specific view of the target object. Additionally, color and depth statistics belonging to the foreground and background are stored as a part of the first level abstraction.

Second level abstraction on the other hand is a *pre-image*, $\Theta_t = \{\mathbf{P}_t, \mathbf{R}_t, \mathbf{T}_t\}$ where $\mathbf{P}_t$ is the set of surfels modelling the 3D target in target's coordinate system while $\{\mathbf{R}_t, \mathbf{T}_t\}$ represents the target's position in camera coordinate system.

The interaction between the two is defined as follows; the first level abstraction localizes the target and extracts a foreground region. This region is then used for updating the second level abstraction by attempting to update the 3D model via ICP [109]. In case of a successful 3D model update, it is projected back onto the 2D image plane as the spatial support for the DCF update. Afterwards, in case a significant aspect ratio change is detected in the projected region compared to the reference region, currently used filter is added to $\{\mathbf{h^s}\}_{s=1}^{S}$. On every N^{th} frame, all filters in $\{\mathbf{h^s}\}_{s=1}^{S}$ are evaluated on the given image and if another filter $\mathbf{h}^s$ than the one in use $\mathbf{h}^t$ produces the maximum correlation filter, $\mathbf{h}^s$ is taken into use. This approach ensures the usage of correct filter for possible view changes and minimizes the possibility of false positive/negative occlusions, target model corruptions etc. An overview of the approach is given in Figure. 4.4.

The proposed approach is evaluated on two widely used benchmarks; the first one is the PTB [113] with the results provided in Table 4.3 and the second one is STC [135] with the results provided in Table 4.4.

The results clearly show that the proposed approach is superior and provides state-of-the-art performance on both benchmarks. One category where the proposed approach struggles is *Animal* where non-rigid deformations is common. However, this is expected since the 3D pre-image cannot handle those cases well.

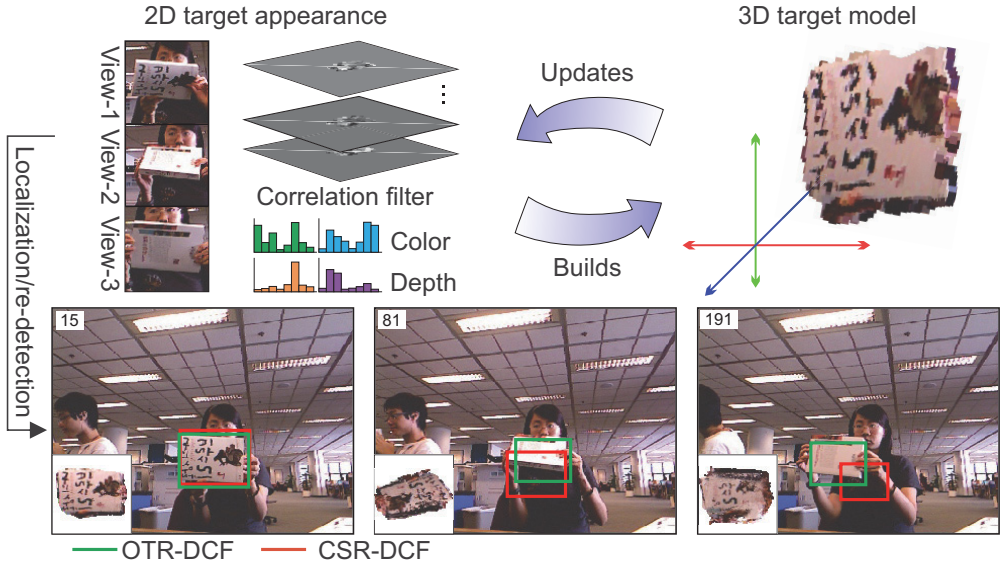

Figure 4.4 The overview of the proposed algorithm in [P3].

Table 4.3 Experiments on the Princeton Tracking Benchmark using the PTB protocol. Numbers in the parenthesis are the ranks [P3].

Method	Avg Rank	Tracking Category										
		Human	*Animal*	*Rigid*	*Large*	*Small*	*Slow*	*Fast*	*Occ.*	*No-Occ.*	*Passive*	*Active*
OTR	**2.36**	0.77(2)	0.68(6)	0.81(2)	0.76(4)	**0.77(1)**	0.81(2)	**0.75(1)**	0.71(3)	0.85(2)	**0.85(1)**	0.74(2)
ca3dms+toh [72]	4.55	0.66(9)	0.74(2)	**0.82(1)**	0.73(7)	0.74(2)	0.80(4)	0.71(7)	0.63(9)	**0.88(1)**	0.83(2)	0.70(6)
CSR-rgbd++ [57]	5.00	0.77(3)	0.65(8)	0.76(7)	0.75(5)	0.73(3)	0.80(3)	0.72(4)	0.70(4)	0.79(8)	0.79(6)	0.72(4)
3D-T [11]	5.64	**0.81(1)**	0.64(9)	0.73(12)	**0.80(1)**	0.71(6)	0.75(9)	0.75(2)	**0.73(1)**	0.78(11)	0.79(7)	0.73(3)
PT [113]	6.09	0.74(6)	0.63(11)	0.78(3)	0.78(3)	0.70(7)	0.76(5)	0.72(6)	0.72(2)	0.75(13)	0.82(4)	0.70(7)
OAPF [92]	6.09	0.64(12)	**0.85(1)**	0.77(6)	0.73(8)	0.73(5)	**0.85(1)**	0.68(9)	0.64(8)	0.85(3)	0.78(9)	0.71(5)
DLST [2]	6.45	0.77(4)	0.69(5)	0.73(13)	0.80(2)	0.70(9)	0.73(11)	0.74(3)	0.66(6)	0.85(4)	0.72(13)	**0.75(1)**
DM-DCF [58]	6.91	0.76(5)	0.58(13)	0.77(5)	0.72(9)	0.73(4)	0.75(8)	0.72(5)	0.69(5)	0.78(10)	0.82(3)	0.69(9)
DS-KCF-Shape [43]	7.27	0.71(7)	0.71(4)	0.74(9)	0.74(6)	0.70(8)	0.76(6)	0.70(8)	0.65(7)	0.81(6)	0.77(11)	0.70(8)
DS-KCF [86]	9.91	0.67(8)	0.61(12)	0.76(8)	0.69(10)	0.70(10)	0.75(10)	0.67(11)	0.63(10)	0.78(12)	0.79(8)	0.66(10)
DS-KCF-CPP [43]	10.09	0.65(10)	0.64(10)	0.74(10)	0.66(12)	0.69(12)	0.76(7)	0.65(12)	0.60(12)	0.79(9)	0.80(5)	0.64(12)
hiob-lc2 [115]	10.18	0.53(13)	0.72(3)	0.78(4)	0.61(13)	0.70(11)	0.72(12)	0.64(13)	0.53(13)	0.85(5)	0.77(12)	0.62(13)
STC [135]	10.45	0.65(11)	0.67(7)	0.74(11)	0.68(11)	0.69(13)	0.72(13)	0.68(10)	0.61(11)	0.80(7)	0.78(10)	0.66(11)

Table 4.4 The normalized area under the curve (AUC) scores computed from one-pass evaluation on the STC Benchmark [135] [P3].

	Attributes										
Method	*AUC*	*IV*	*DV*	*SV*	*CDV*	*DDV*	*SDC*	*SCC*	*BCC*	*BSC*	*PO*
OTR	**0.49**	**0.39**	**0.48**	**0.31**	0.19	**0.45**	**0.44**	0.46	**0.42**	**0.42**	**0.50**
CSR-rgbd++ [57]	0.45	0.35	0.43	0.30	0.14	0.39	0.40	0.43	0.38	0.40	0.46
ca3dms+toh [72]	0.43	0.25	0.39	0.29	0.17	0.33	0.41	**0.48**	0.35	0.39	0.44
STC [135]	0.40	0.28	0.36	0.24	**0.24**	0.36	0.38	0.45	0.32	0.34	0.37
DS-KCF-Shape [43]	0.39	0.29	0.38	0.21	0.04	0.25	0.38	0.47	0.27	0.31	0.37
PT [113]	0.35	0.20	0.32	0.13	0.02	0.17	0.32	0.39	0.27	0.27	0.30
DS-KCF [86]	0.34	0.26	0.34	0.16	0.07	0.20	0.38	0.39	0.23	0.25	0.29
OAPF [92]	0.26	0.15	0.21	0.15	0.15	0.18	0.24	0.29	0.18	0.23	0.28

4.4 P4 CDTB: A Color and Depth Visual Object Tracking Dataset and Benchmark

A major problem of generic visual object tracking in RGBD has been the lack of available datasets to evaluate the algorithms. To the best of our knowledge, only PTB [113] and STC [135] fit to this task. In spite of their usefulness in the early RGBD tracking algorithm development, these aforementioned benchmarks have their shortcomings which we believe hamper the RGBD tracker development. Some of these issues can be named as the diversity of the target objects, sensors and scenarios (*e.g.* long-term occlusions, out of frame occlusions etc.). Thus, it gradually has become clear that a more sophisticated and diverse RGBD dataset was necessary to address these shortcomings for facilitating further RGBD tracking improvements.

In [P4], we propose a novel RGBD tracking benchmark with the visual object tracking community's requirements in mind. Our contributions are as follows;

- Usage of multiple sensors with different modalities consisting of active and passive sensors
- Including outdoors sequences as well as indoor sequences to evaluate the effectiveness of the algorithms in different environments
- Capturing sequences with significant object pose changes for challenging the algorithms to their maximum
- Having sequences with long occlusions and out of view scenarios
- Comparison of state-of-the-art RGB, their RGBD extensions and native RGBD trackers

For capturing an extensive set of sequences with different properties, three different setups have been used.

The first setup is a Kinect v2 which outputs a 24-bit 1920 x 1080 RGB images along with 512 x 512 32-bit floating point depth images at 30 FPS.

The second setup is a Time-of-Flight (ToF) - RGB pair with Basler tof640-20gm ToF and Basler acA1920-50gc color camera. ToF camera provides 640 x 480 pixels resolution at maximum 20 FPS and the color camera outputs 1920 x 1200 pixels at maximum 50 FPS with an external synchronisation used as trigger. This setup was mounted on a CNC machined aluminum base to avoid calibration errors.

Table 4.5 Comparison of CDTB with related benchmarks in the number of RGBD devices used for acquisition (N_{HW}), presence of indoor and outdoor sequences (In/Out), per-frame attribute annotation (Per-frame), number of attributes (N_{atr}), number of sequences (N_{seq}), total number of frames (N_{frm}) average sequence length (N_{avg}), number of frames with target not visible (N_{out}), number of target disappearances (N_{dis}), average length of target absence period (N_{avgout}), number of times a target rotates away from the camera by at least 180°(N_{rot}), average number of target rotations per sequence (N_{seqrot}) and tracking performance under the PTB protocol ($\Omega_{0.5}$) [P4].

Dataset	N_{HW}	In	Out	Per-frame	N_{atr}	N_{seq}	N_{frm}	N_{avg}	N_{out}	N_{avgout}	N_{dis}	N_{rot}	N_{seqrot}	$\Omega_{0.5}$
CDTB	3	✓	✓	✓	13	80	101,956	1,274	10,656	56.4	189	358	4.5	0.316
STC [135]	1	✓	✓	✓	12	36	9,195	255	0	0	0	30	0.8	0.530
PTB [113]	1	✓	✗	✗	5	95	20,332	214	846	6.3	134	83	0.9	0.749

The last setup is a stereo Basler acA1920-50gc color camera pair. This setup enabled extraction of depth information in outdoors sequences without any degradation due to the sunlight contrary to the active depth sensor setups. It is a significant difference than the previous benchmarks since none of them includes this type of sequences to the best of our knowledge. Example images from each capturing setup is given in Figure 4.5.

Our benchmark was manually annotated using VOT Aibu annotation tool [1] with axis-aligned bounding boxes by adopting the VOT [65] definition. A summary for the features of the proposed benchmark is given in Table 4.5.

In order to validate the benchmark, 16 trackers in three categories were evaluated; (i) state-of-the-art short-term RGB trackers (KCF [48], NCC [66], BACF [59], CSRDCF [80], SiamFC [10], ECOhc [31], ECO [31] and MDNet [97]), (ii) state-of-the-art long-term RGB trackers; (TLD [55], FuCoLoT [81] and MBMD [137]) and (iii) state-of-the-art native RGBD trackers; (OTR [123] and Ca3dMS [72]). Similar to [P2], RGBD augmented versions of a few state-of-the-art RGB trackers; ECOhc-D [57], CSRDCF-D [57] and KCF-D[2] are also included.

An important finding of the experiments is the fact that the top performers, MDNet [97], MBMD [137], were both RGB trackers with strong re-detection capabilities. Even though MDNet was proposed as a short-term tracker, its CNN-based classifier and selective update mechanism equipped it with long-term capabilities. This led to the conclusion that despite they lack the extra information from the depth

[1] https://github.com/votchallenge/aibu
[2] Uses depth as a feature channel

channel, they are able to compensate this using the deep features since the state-of-the-art native RGBD trackers OTR [123] and CSRDCF-D [57] use hand crafted features such as HOG [95] and color names [127].

Overall, this paper provided multiple important cues for the development of RGBD trackers.

- Using deep networks on depth channel may provide significant performance gain as they are able to capture appearance changes better than the hand crafted features
- Depth information provides strong clues for successful failure detection
- Adding depth information complements traditional RGB based approaches as it provides important cues regarding 3D target appearance and foreground detection

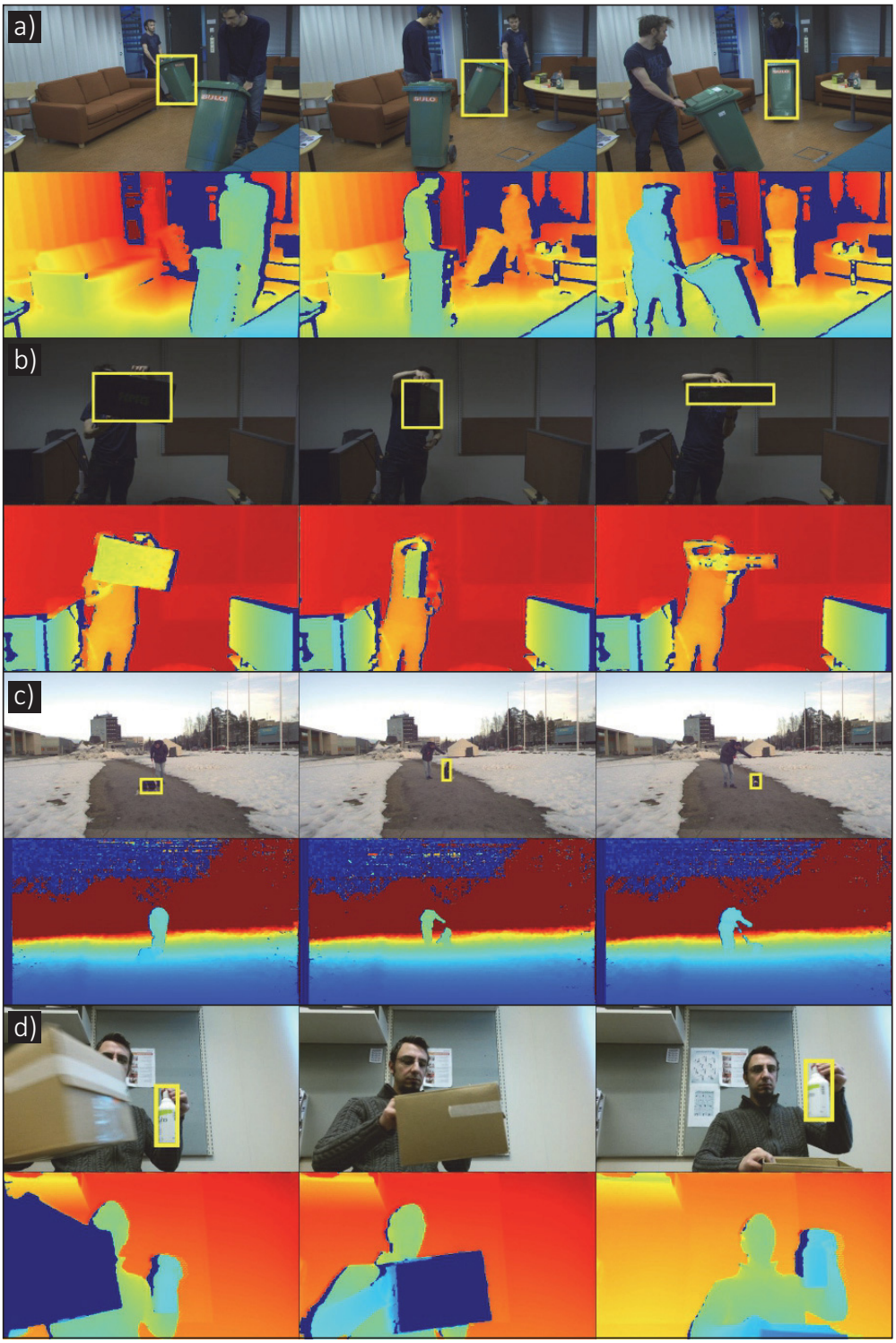

Figure 4.5 Examples of images captured with different hardware setups. a and b are from the ToF-RGB pair, (c) the stereo-camera sensor and (d) from the Kinect sensor [P4].

5 CONCLUSIONS

Generic visual object tracking aims to provide efficient solutions to a fundamental computer vision problem which is to localize a target object in videos with its location known only in the first frame. Having many real-life applications such as augmented reality, surveillance, sports, robotics etc. makes it an attractive research field and the last few years have seen a surge in the number of proposed algorithms. Thanks to the easy and cheap accessibility of the depth information in recent years, it is intuitive to use the additional depth information to further improve the tracker performances by coping with long-term occlusions and appearance changes. With that in mind, this thesis proposes three novel, generic visual RGBD tracking algorithms [P1, P2, P3] and a novel RGBD tracking benchmark [P4].

The first proposed method [P1] adds the depth information into a spatially constrained DCF formulation. The experiments conducted on a well-known RGBD tracking benchmark showed that using the depth channel indeed significantly increases the tracker performance compared to the baseline tracker while obtaining on par results with the state-of-the-art at a fraction of its computational complexity. However, lack of color information in the mask generation stage prohibits further performance boost.

The second proposed method [P2] addressed the mask generation issues in [P1] and offered a generic, two-stage framework for converting any short-term RGB tracker into an RGBD tracker with re-detection capabilities. The results of the experiments validated the effectiveness of the depth channel by consistently improving the tracker performances in all RGB trackers while the full integration (two-stage) of the framework outperformed the state-of-the-art by a large margin on the day of its submission. However, the algorithm proposed in [P2] was not able to cope with certain appearance changes such as out-of-plane rotation since it still models the target object on a 2D plane.

To cope with out-of-plane rotations, the third proposed method [P3] fused SLAM

approach with spatially constrained DCF formulation by using the reprojections of a selectively updated 3D model as a spatial constraint and a cue for creating a set of target views to alleviate the issues related to the 2D plane based target modelling. This approach outperformed the state-of-the-art on both well-known RGBD datasets on the day of its submission and showed the efficiency of the proposed method.

During the development of [P1], [P2] and [P3], it was seen that the existing RGBD datasets were insufficient to facilitate further RGBD tracking development. This is mainly due to the lack of different sensor modalities and the diversity of the scenarios. In order to address these issues, a novel RGBD tracking dataset was proposed in [P4]. Extensive experiments using RGB, native RGBD and depth augmented RGBD trackers showed that despite the significant improvements in RGBD trackers in recent years, they still lag behind the state-of-the-art RGB trackers. This can be attributed to the fact that the state-of-the-art RGB trackers use deep features which have superior discriminative capabilities whereas RGBD trackers still use hand crafted features. Thus, [P4] offers an important insight for future RGBD tracker development and makes a valuable addition to the tracking community with a novel dataset.

In conclusion, this thesis contributes to the RGBD tracking field with three novel trackers with two of them achieving state-of-the-art results on the day of their submissions. The experiments show that the depth channel provides important cues and improves the results significantly. However, it is clear that the future RGBD trackers have to adopt deep networks for feature generation since they learn the target appearance significantly better than the hand crafted features.

REFERENCES

[1] A. Adam, E. Rivlin and I. Shimshoni. Robust Fragments-based Tracking Using the Integral Histogram. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[2] N. An, X.-G. Zhao and Z.-G. Hou. Online RGB-D Tracking via Detection-Learning-Segmentation. *International Conference on Pattern Recognition (ICPR)*. 2016.

[3] B. Babenko, M. Yang and S. Belongie. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011), 1619–1632.

[4] S. Bae and K. Yoon. Robust Online Multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[5] R. Bellman. A Markovian Decision Process. *Journal of Mathematics and Mechanics* 6.5 (1957), 679–684.

[6] J. Berclaz, E. Turetken, F. Fleuret and P. Fua. Multiple Object Tracking using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (2011), 1806–1819.

[7] P. Bergmann, T. Meinhardt and L. Leal-Taixé. Tracking Without Bells and Whistles. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[8] K. Bernardin and R. Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing* (Jan. 2008).

[9] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik and P. H. S. Torr. Staple: Complementary Learners for Real-Time Tracking. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2016.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. S. Torr. Fully-Convolutional Siamese Networks for Object Tracking. *European Conference on Computer Vision (ECCV) Workshops*. 2016.

[11] A. Bibi, T. Zhang and B. Ghanem. 3D Part-Based Sparse Tracker with Automatic Synchronization and Registration. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[12] M. B. Blaschko and C. H. Lampert. Learning to Localize Objects with Structured Output Regression. *European Conference on Computer Vision (ECCV)*. 2008.

[13] D. S. Bolme, J. Beveridge, B. A. Draper and Y.-M. Lui. Visual Object Tracking using Adaptive Correlation Filters. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[14] A. Bordes, L. Bottou, P. Gallinari and J. Weston. Solving Multiclass Support Vector Machines with LaRank. *International Conference on Machine Learning*. 2007.

[15] A. Bordes, N. Usunier and L. Bottou. Sequence Labelling SVMs Trained in One Pass. *Machine Learning and Knowledge Discovery in Databases*. 2008.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* 3.1 (2011), 1–122.

[17] Y. Boykov, O. Veksler and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.11 (2001).

[18] K. Briechle and U. Haneback. Template Matching Using Fast Normalized Cross Correlation. *SPIE* 4387 (2001), 95–102.

[19] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger and R. Shah. Signature Verification Using a "Siamese" Time Delay Neural Network. *International Conference on Neural Information Processing Systems*. 1993.

[20] T. Brox and J. Malik. Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3 (2011), 500–513.

[21] W. Budiharto, V. Andreas, J. S. Suroso, A. A. S. Gunawan and E. Irwansyah. Development of Tank-Based Military Robot and Object Tracker. *Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. 2019, 221–224.

[22] L. Čehovin, M. Kristan and A. Leonardis. An Adaptive Coupled-layer Visual Model for Robust Visual Tracking. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2011.

[23] H. Chang, M. Park, H. Jeong and J. Choi. Tracking Failure Detection by Imitating Human Visual Perception. *International Conference on Image Processing (ICIP)*. 2011.

[24] V. Chari, S. Lacoste-Julien, I. Laptev and J. Sivic. On Pairwise Costs for Network Flow Multi-Object Tracking. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[25] F. Chaumette, P. Rives and B. Espiau. Positioning of a Robot with Respect to an Object, Tracking It and Estimating Its Velocity by Visual Servoing. *IEEE International Conference on Robotics and Automation (ICRA)*. 1991.

[26] R. Collins, X. Zhou and S. Teh. An Open Source Tracking Testbed and Evaluation Web Site. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance* (2005).

[27] D. Comaniciu, V. Ramesh and P. Meer. Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), 564–567.

[28] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning* 20.3 (1995), 273–297.

[29] J. Dai, Y. Li, K. He and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *Advances in Neural Information Processing Systems 29*. 2016, 379–387.

[30] M. Danelljan, G. Hager, F. S. Khan and M. Felsberg. Discriminative Scale Space Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.8 (2017), 1561–1575.

[31] M. Danelljan, G. Bhat, F. Shahbaz Khan and M. Felsberg. ECO: Efficient Convolution Operators for Tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[32] M. Danelljan, A. Robinson, F. Shahbaz Khan and M. Felsberg. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. *European Conference on Computer Vision (ECCV)*. 2016.

[33] A. Diplaros, N. Vlassis and T. Gevers. A Spatially Constrained Generative Model and an EM Algorithm for Image Segmentation. *IEEE Transactions on Neural Networks* 18.3 (2007).

[34] M. Everingham, L. Gool, C. K. Williams, J. Winn and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88.2 (June 2010), 303–338.

[35] H. Fan and H. Ling. Parallel Tracking and Verifying: A Framework for Real-Time and High Accuracy Visual Tracking. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[36] C. Feichtenhofer, A. Pinz and A. Zisserman. Detect to Track and Track to Detect. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[37] J. Ferryman and A. Shahrokni. PETS2009: Dataset and Challenge. *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. 2009, 1–6.

[38] R. B. Fisher. The PETS04 Surveillance Ground-truth Data Sets. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance* (2004).

[39] H. K. Galoogahi, T. Sim and S. Lucey. Multi-channel Correlation Filters. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2013.

[40] H. Galoogahi, T. Sim and S. Lucey. Correlation Filters with Limited Boundaries. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[41] H. Grabner, C. Leistner and H. Bischof. Semi-Supervised On-line Boosting for Robust Tracking. *European Conference on Computer Vision (ECCV)*. 2008.

[42] R. W. Hamming. Error Detecting and Error Correcting Codes. *The Bell System Technical Journal* 29.2 (1950), 147–160.

[43] S. Hannuna, M. Camplani, J. Hall, M. Mirmehdi, D. Damen, T. Burghardt, A. Paiement and L. Tao. DS-KCF: A Real-time Tracker for RGB-D Data. *Journal of Real-Time Image Processing* (2016).

[44] S. Hare, A. Saffari and P. H. S. Torr. Struck: Structured Output Tracking with Kernels. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2011.

[45] K. He, G. Gkioxari, P. Dollár and R. Girshick. Mask R-CNN. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[46] K. He, X. Zhang, S. Ren and J. Sun. Deep Residual Learning for Image Recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[47] D. Held, S. Thrun and S. Savarese. Learning to Track at 100 FPS with Deep Regression Networks. *European Conference Computer Vision (ECCV)*. 2016.

[48] J. F. Henriques, R. Caseiro, P. Martins and J. Batista. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.3 (2015), 583–596.

[49] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of research of the National Bureau of Standards* 49 (1952), 409–436.

[50] Z. Hong, Zhe Chen, C. Wang, X. Mei, D. Prokhorov and D. Tao. MUlti-Store Tracker (MUSTer): A Cognitive Psychology Inspired Approach to Object Tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[51] B. K. P. Horn and B. G. Schunck. Determining Optical Flow. *Artificial Intelligence* 17.1–3 (Aug. 1981), 185–203.

[52] D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge. Comparing Images Using the Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9 (1993), 850–863.

[53] A. E. Johnson. *Spin-images: A Representation for 3-d Surface Matching*. Tech. rep. 1997.

[54] Junseok Kwon and Kyoung Mu Lee. Tracking by Sampling Trackers. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2011.

[55] Z. Kalal, K. Mikolajczyk and J. Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.7 (2011), 1409–1422.

[56] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), 35–45.

[57] U. Kart, J.-K. Kämäräinen and J. Matas. How to Make an RGBD Tracker ?: (2018).

[58] U. Kart, J.-K. Kämäräinen, J. Matas, L. Fan and F. Cricri. Depth Masked Discriminative Correlation Filter. (2018).

[59] H. Kiani Galoogahi, A. Fagg, C. Huang, D. Ramanan and S. Lucey. Need for Speed: A Benchmark for Higher Frame Rate Object Tracking. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[60] C. Kim, F. Li, A. Ciptadi and J. M. Rehg. Multiple Hypothesis Tracking Revisited. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015.

[61] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukežič, A. Garcia-Martin, A. Saffari, A. Petrosino and A. Solis Montero. The Visual Object Tracking VOT2015 Challenge Results. *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2015.

[62] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pfugfelder, L. C. Zajc, T. Vojir, G. Bhat, A. Lukežič, A. Eldesokey, G. Fernandez and et al. The Sixth Visual Object Tracking VOT2018 Challenge Results. *European Conference on Computer Vision (ECCV) Workshops*. 2018.

[63] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kämäräinen, L. Č. Zajc, O. Drbohlav, A. Lukežič, A. Berg, A. Eldesokey, J. Kapyla and G. Fernandez. The Seventh Visual Object Tracking VOT2019 Challenge Results. *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2019.

[64] M. Kristan, J. Matas, G. Nebehay, F. Porikli and L. Čehovin. A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.11 (2016), 2137–2155.

[65] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojır and et al. The Visual Object Tracking VOT2014 Challenge Results. *European Conference on Computer Vision (ECCV) Workshops*. 2014.

[66] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli and et al. The Visual Object Tracking VOT2013 Challenge Results. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2013.

[67] J. Kwon, K. M. Lee and F. C. Park. Visual Tracking via Geometric Particle Filtering on the Affine Group with Optimal Importance Functions. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[68] R. Least-squares, R. Rifkin, G. Yeo and T. Poggio. Regularized Least-Squares Classification. *Advances in Learning Theory: Methods, Model and Applications, NATO Science Series III: Computer and Systems Sciences* 190 (June 2003).

[69] Li Zhang, Yuan Li and R. Nevatia. Global Data Association for Multi-object Tracking Using Network Flows. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[70] H. Li, C. Shen and Q. Shi. Real-time Visual Tracking Using Compressive Sensing. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[71] H. Liu, X. Yang, L. J. Latecki and S. Yan. Dense Neighborhoods on Affinity Graph. *International Journal of Computer Vision* 98 (2011), 65–82.

[72] Y. Liu, X. Jing, J. Nie, H. Gao, J. Liu and G. Jiang. Context-Aware Three-Dimensional Mean-Shift With Occlusion Handling for Robust Object Tracking in RGB-D Videos. *IEEE Transactions on Multimedia* 21.3 (2019), 664–677.

[73] B. P. L. Lo and S. A. Velastin. Automatic Congestion Detection System for Underground Platforms. *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001*. 2001, 158–161.

[74] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60 (2004), 91–110.

[75] W. Lu, J. 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 (2013), 1704–1716.

[76] M. Luber, L. Spinello and K. O. Arras. People Tracking in RGB-D Data with On-line Boosted Target Models. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, 3844–3849.

[77] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *IJCAI*. 1981.

[78] A. Lukežič, U. Kart, J. Käpylä, A. Durmush, J.-K. Kämäräinen, J. Matas and M. Kristan. CDTB: A Color and Depth Visual Object Tracking Dataset and Benchmark. (2019).

[79] A. Lukežič, L. Čehovin Zajc, T. Vojíř, J. Matas and M. Kristan. Now You See Me: Evaluating Performance in Long-term Visual Tracking. (Apr. 2018).

[80] A. Lukežič, T. Vojır, L. Čehovin, J. Matas and M. Kristan. Discriminative Correlation Filter with Channel and Spatial Reliability. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[81] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas and M. Kristan. FuCoLoT – A Fully-Correlational Long-Term Tracker. *Asian Conference on Computer Vision (ACCV)*. 2018.

[82] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao and T. Kim. Multiple Object Tracking: A Literature Review. *arXiv: Computer Vision and Pattern Recognition* (2014).

[83] C. Ma, J.-B. Huang, X. Yang and M.-H. Yang. Adaptive Correlation Filters with Long-Term and Short-Term Memory for Object Tracking. *International Journal of Computer Vision* 126 (2018), 771–796.

[84] C. Ma, X. Yang, C. Zhang and M.-H. Yang. Long-Term Correlation Tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[85] A. Maksai, X. Wang, F. Fleuret and P. Fua. Non-Markovian Globally Consistent Multi-Object Tracking. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[86] Massimo Camplani, Sion Hannuna, Majid Mirmehdi, Dima Damen, Adeline Paiement, Lili Tao and Tilo Burghardt. Real-time RGB-D Tracking with Depth Scaling Kernelised Correlation Filters and Occlusion Handling. *British Machine Vision Conferennce (BMVC)*. 2015.

[87] Matej Kristan, Ales Leonardis, Jiř'i Matas, Michael Felsberg, Roman Pflugfelder and et al. The Visual Object Tracking VOT2017 Challenge Results. *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2017.

[88] Matej Kristan, Ales Leonardis, Jiř'i Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomas Vojır and et al. The Visual Object Tracking VOT2016 Challenge Results. *European Conference on Computer Vision (ECCV) Workshops*. 2016.

[89] X. Mei and H. Ling. Robust Visual Tracking Using L1 Minimization. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2009.

[90] X. Mei, H. Ling, Y. Wu, E. Blasch and L. Bai. Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[91] K. Meshgi, S.-i. Maeda, S. Oba and S. Ishii. Fusion of Multiple Cues from Color and Depth Domains Using Occlusion Aware Bayesian Tracker. *IEICE Tech. Rep. Neurocomp* 113 (Jan. 2014), 127–132.

[92] K. Meshgi, S.-i. Maeda, S. Oba, H. Skibbe, Y.-z. Li and S. Ishii. An Occlusion-aware Particle Filter Tracker to Handle Complex and Persistent Occlusions. *Computer Vision and Image Understanding* 150 (2016), 81–94.

[93] A. Milan, S. Roth and K. Schindler. Continuous Energy Minimization for Multitarget Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.1 (2014), 58–72.

[94] M. Mueller, N. Smith and B. Ghanem. A Benchmark and Simulator for UAV Tracking. *European Conference on Computer Vision (ECCV)*. 2016.

[95] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[96] M. A. Naiel, M. O. Ahmad, M. Swamy, J. Lim and M.-H. Yang. Online Multi-Object Tracking via Robust Collaborative Model and Sample Selection. *Computer Vision and Image Understanding* 154.C (2017), 94–107.

[97] H. Nam and B. Han. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[98] T. Nawaz and A. Cavallaro. A Protocol for Evaluating Video Trackers Under Real-World Conditions. *IEEE Transactions on Image Processing* 22.4 (2013), 1354–1361.

[99] H. T. Nguyen and A. W. M. Smeulders. Fast Occluded Object Tracking by a Robust Appearance Filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.8 (2004), 1099–1104.

[100] T. Ojala, M. Pietikainen and T. Maenpaa. Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (2002), 971–987.

[101] S. Oron, A. Bar-Hillel, D. Levi and S. Avidan. Locally Orderless Tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[102] M. Ozuysal, P. Fua and V. Lepetit. Fast Keypoint Recognition in Ten Lines of Code. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[103] F. Pernici and A. Del Bimbo. Object Tracking by Oversampling Local Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.12 (2014), 2538–2551.

[104] S. Ren, K. He, R. Girshick and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), 1137–1149.

[105] E. Ristani and C. Tomasi. Tracking Multiple People Online and in Real Time. *Asian Conference on Computer Vision (ACCV)*. 2014.

[106] A. Roshan Zamir, A. Dehghan and M. Shah. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. *European Conference on Computer Vision (ECCV)*. 2012.

[107] D. A. Ross, J. Lim, R.-S. Lin and M.-H. Yang. Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision* 77 (2008), 125–141.

[108] Y. Rubner, C. Tomasi and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* 40.2 (2000), 99–121.

[109] M. Rünz and L. Agapito. Co-Fusion: Real-time Segmentation, Tracking and Fusion of Multiple Objects. *ICRA*. 2017.

[110] 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* 115.3 (2015), 211–252.

[111] J. Sivic and A. Zisserman. Efficient Visual Search of Videos Cast as Text Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.4 (2009), 591–606.

[112] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan and M. Shah. Visual Tracking: An Experimental Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (2014), 1442–1468.

[113] S. Song and J. Xiao. Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2013.

[114] L. Spinello and K. O. Arras. People Detection in RGB-D Data. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, 3838–3843.

[115] P. Springstübe, S. Heinrich and S. Wermter. Continuous Convolutional Object Tracking. *ESANN*. 2018.

[116] Suha Kwak, Woonhyun Nam, Bohyung Han and Joon Hee Han. Learning Occlusion with Likelihoods for Visual Tracking. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2011.

[117] J. W. Suurballe. Disjoint Paths in a Network. *Networks* 4.2 (1974), 125–145.

[118] Y. Taigman, M. Yang, M. Ranzato and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[119] M. E. Tipping and C. M. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society Series B* 61.3 (1999), 611–622.

[120] P. H. S. Torr and A. Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding* 78 (2000), 138–156.

[121] Tse-Wei Chen, Yi-Ling Chen and Shao-Yi Chien. Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space. *2008 IEEE 10th Workshop on Multimedia Signal Processing*. 2008, 322–325.

[122] I. Tsochantaridis, T. Joachims, T. Hofmann and Y. Altun. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.* 6 (Dec. 2005), 1453–1484.

[123] U.Kart, A. Lukežič, M. Kristan, J.-K. Kämäräinen and J. Matas. Object Tracking by Reconstruction with View-Specific Discriminative Correlation Filters. (2019).

[124] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi and P. H. S. Torr. End-To-End Representation Learning for Correlation Filter Based Tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[125] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger and B. Leibe. MOTS: Multi-Object Tracking and Segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[126] C. Wang, Y. Wang, Y. Wang, C.-T. Wu and G. Yu. muSSP: Efficient Min-cost Flow Algorithm for Multi-object Tracking. *Advances in Neural Information Processing Systems 32*. 2019, 425–434.

[127] J. van de Weijer, C. Schmid, J. Verbeek and D. Larlus. Learning Color Names for Real-world Applications. *IEEE Transactions on Image Processing* 18.7 (2009), 1512–1523.

[128] Weiming Hu, Tieniu Tan, Liang Wang and S. Maybank. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 34.3 (2004), 334–352.

[129] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi and S. Z. Li. Multiple Target Tracking Based on Undirected Hierarchical Relation Hypergraph. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[130] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2 (2009), 210–227.

[131] Y. Wu, J. Lim and M. Yang. Online Object Tracking: A Benchmark. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[132] Y. Wu, J. Lim and M. Yang. Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9 (2015), 1834–1848.

[133] Y. Wu, J. Lim and Y. Ming-Hsuan. Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (2015), 1834–1848.

[134] Y. Xiang, A. Alahi and S. Savarese. Learning to Track: Online Multi-Object Tracking by Decision Making. *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015.

[135] J. Xiao, R. Stolkin, Y. Gao and A. Leonardis. Robust Fusion of Color and Depth Data for RGB-D Target Tracking Using Adaptive Range-Invariant Depth Models and Spatio-Temporal Consistency Constraints. *IEEE Transactions on Cybernetics* 48 (2018), 2485–2499.

[136] S. Zagoruyko and N. Komodakis. Learning to Compare Image Patches via Convolutional Neural Networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[137] Y. Zhang, D. Wang, L. Wang, J. Qi and H. Lu. Learning Regression and Verification Networks for Long-term Visual Tracking. *CoRR* abs/1809.04320 (2018).

[138] Z. Zhang. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *International Journal of Computer Vision* 13.2 (Oct. 1994), 119–152.

[139] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang and M.-H. Yang. Online Multi-Object Tracking with Dual Matching Attention Networks. *European Conference on Computer Vision (ECCV)*. 2018.

PUBLICATIONS

PUBLICATION

I

Depth Masked Discriminative Correlation Filter

U. Kart, J.-K. Kämäräinen, J. Matas, L. Fan and F. Cricri

*International Conference on Pattern Recognition (ICPR)*2018

Depth Masked Discriminative Correlation Filter

Uğur Kart*, Joni-Kristian Kämäräinen*
*Laboratory of Signal Processing
Tampere University of Technology
Email: first.surname@tut.fi

Jiří Matas†
†Faculty of Electrical Engineering
Czech Technical University in Prague
Email: matas@cmp.felk.cvut.cz

Lixin Fan‡, Francesco Cricri‡
‡Nokia Technologies
Tampere, Finland
Email: first.surname@nokia.com

***Abstract*—Depth information provides a strong cue for occlusion detection and handling, but has been largely omitted in generic object tracking until recently due to lack of suitable benchmark datasets and applications. In this work, we propose a *Depth Masked Discriminative Correlation Filter* (DM-DCF) which adopts novel depth segmentation based occlusion detection that stops correlation filter updating and depth masking which adaptively adjusts the spatial support for correlation filter. In Princeton RGBD Tracking Benchmark, our DM-DCF is among the state-of-the-art in overall ranking and the winner on multiple categories. Moreover, since it is based on DCF, "DM-DCF" runs an order of magnitude faster than its competitors making it suitable for time constrained applications.**

I. Introduction

Generic short-term object tracking has been a popular research topic in computer vision for the last few years and trackers based on the *Discriminative Correlation Filter (DCF)* [12] have been particularly successful for applications with time constraint [24], [11], [7], [15]. However, in RGB tracking, there are fundamental difficulties that can be solved with the help of depth (D) information, occlusion handling being the most obvious. Additionally, RGBD sensors are popular in robotics where 3D object tracking also has many important applications, e.g., object manipulation and grasping.

There have been surprisingly small number of works on RGBD tracking since the introduction of the first dedicated benchmark for RGBD tracking: Princeton RGBD Tracking dataset [21]. The dataset authors also proposed baseline trackers for 2D (depth as an additional cue) and 3D tracking (3D bounding box). More recently, particle filter based methods have been proposed by Meshgi *et al.* [16] and Bibi *et al.* [2], but they are both slow for real-time applications. Instead, we adopt the Discriminative Correlation Filter (DCF) approach since it is proven to be fast and successful in RGB tracking. The two other DCF based RGBD works to our best knowledge are Hannuna et al. [10] and An et al. [1] which are both among the top performers in the Princeton dataset.

In aforementioned RGBD tracking methods, depth has been used as a mere additional cue for tracking, but intuitively the most important role for the depth information is in accurate and robust occlusion handling. In our work, instead of separating occlusion handling and tracking, we unite the two processes by integrating occlusion handling with correlation filters in the sense that the spatial filter supports are dynamically altered using the depth based segmentation. In this work we make the following novel contributions:

- We propose a novel occlusion handling mechanism based on object's depth distribution and DCF's response history. By detecting strong occlusions, our tracker avoids corrupting the object model.
- We propose depth masking for DCF which avoids using the occluded regions in matching and therefore provides more reliable tracking scores.
- We experimentally validate Depth Masked Discriminative Correlation Filter (DM-DCF) tracker on the Princeton RGBD Tracking benchmark where it ranks among the state-of-the-art algorithms with ranking first in multiple categories while being faster than the other top performing methods in the benchmark.

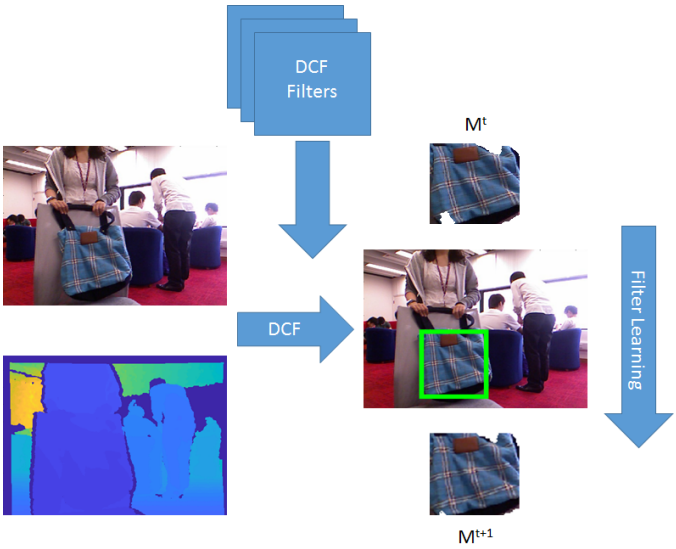

Fig. 1. Overview of our DM-DCF method. Depth cue is used in parallel with DCF tracking by inherently adopting depth based masks into DCF formulation which allows altering the filter supports. Moreover, combination of DCF and depth segmentation mitigates the possible errors stemming from individual observations. (M^t and M^{t+1} are the masks used for filter training where the slight differences between the masks of two consecutive frames are visible)

*Uğur Kart was supported by two projects: Business Finland Project "360 Video Intelligence - For Research Benefit" with Nokia, Lynx, JJ-Net, BigHill, Leonidas and Business Finland - FiDiPro Project "Pocket - Sized Big Visual Data"

†Jiří Matas was supported by Czech Science Foundation Project GACR P103/12/G084.

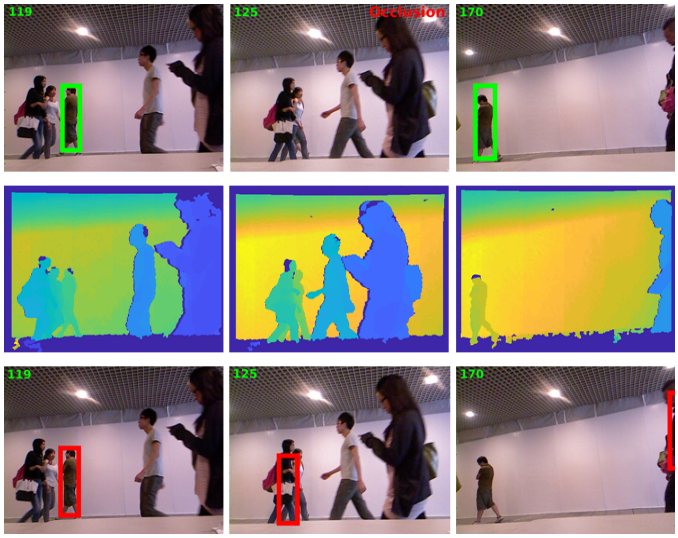

Fig. 2. Example results of our DM-DCF tracker (top row) that uses depth cue (middle row) to construct depth masks. Depth helps detecting occlusions as the third dimension provides a strong clue and hence, the tracker stops updating the object model. However, RGB tracker (CSR-DCF [15], bottom row) is not able to detect the occlusions and continues learning occluding object and consequently, loses the target object.

Our code will be made publicly available to facilitate fair comparisons.

II. Related Work

Object Tracking – Existing trackers for generic short-term object tracking on RGB videos can be grouped under two main categories, *Generative* (matching with updated target model) or *Discriminative* (classifier based) methods, depending on how a tracked object (target region) is modelled and localized [20]. A generative tracker represents the target as a collection of features from previously tracked frames and matches them to search region in the current frame. A few prominent examples of this family of trackers are Incremental Visual Tracking (IVT) [19], Structural Sparse Tracking (SST) [23] and kernel-based object tracking (mean shift) [5]. Generative trackers build a model from the positive examples, *e.g*. tracked regions, but false matches occur if the background or other objects have similar texture properties. This issue is addressed in discriminative trackers by continuously learning and updating a classifier that distinguishes the target from background. In recent years, discriminative approach has been more popular and it has produced many well-performing trackers such as Tracking-Learning-Detection (TLD) [13], Continuous Convolutional Operators Tracking (C-COT) [8], Multi-Domain Convolutional Neural Network (MDNet) [17], ECO [7], CSR-DCF [15]. Due to its mathematical simplicity, efficiency and superior performance, we adopted the DCF based approach as our baseline to allow us to achieve fast throughput rate while retaining an accuracy comparable to more complicated algorithms as it is proven in VOT 2017 [24].

Object Tracking with Depth – The number of research papers on generic RGBD (color + depth) object tracking is surprisingly limited despite the fact that depth sensors are ubiquitous and the apparent application of the problem in robotics. One of the reasons is the lack of suitable datasets until recent years. In 2013 Song *et al*. [21] introduced the Princeton Dataset for RGBD tracking and nine variations of a baseline tracker. They presented two main approaches; *depth as an additional cue* and *point cloud tracking*. In the first case, depth is added as an additional dimension to Histogram of Oriented Gradients (HOG) feature space and in the second case tracking is based on 3D point clouds producing also a 3D bounding box. Their best method performs well, but contains heuristic processing stages and its speed (0.26 FPS) makes the algorithm unsuitable for real-time applications.

Another RGBD method was recently proposed by Meshgi *et al*. [16] who tackled the problem with an occlusion-aware particle filter framework employing a probabilistic model. Although their algorithm is among the top performers on Princeton Benchmark, the complexity of their model, the number of parameters to be tuned and the slowness of the algorithm (0.9 FPS) makes it unpractical for many applications.

Bibi *et al*. [2] suggested a part-based sparse tracker within a particle filter framework. They represented particles as 3D cuboids that consist of sparse linear combination of dictionary elements which are not updated over the time. In case of no occlusion, their method first finds a rough estimate of the target location using 2D optical flow. Following this, particles are sampled in rotation $\boldsymbol{R}$ and translation $\boldsymbol{T}$ space. To detect the occlusions, they adopted a heuristic which states that the number of points in the 3D cuboid of the target should decrease below a threshold. Their method sets currently the state-of-the-art on Princeton benchmark however, computation times are not mentioned in the original work.

To the authors' best knowledge there are only two RGBD trackers based on DCF which is used in our method. The first one was proposed by Hannuna *et al*. [10] who use depth for occlusion handling, scale and shape analysis. To this end, they first apply a clustering scheme on depth histogram which is followed by formation of a single Gaussian distribution based depth modelling where they assume the cluster with the smallest mean must be the object (similar heuristic used in Song *et al*. [21]). Another shortcoming of their algorithm is that their occlusion handling allows tracking of occluding region which introduces same problems as in [21].

Second method was proposed by An *et al*. [1]. They used depth based segmentation in parallel to Kernelized Correlation Filter (KCF) [11] detection and then interpreted the results to locate the target and determine the occlusion state with a heuristic approach.

The rest of the paper is organized as follows; Section III provides an overview for the proposed tracking method, Section IV reports the results of our tracker, its comparison against the state-of-the-art RGBD trackers and also discusses the ablation studies to evaluate the impacts of our design choices. Finally Section V sums up our proposed method with remarks for the future work. The overview of the proposed method can be seen in Fig. 1 and Alg. 1.

III. Depth Masked DCF

Our approach is inspired by the recent work of Lukezic et al. [15] who robustified standard DCF by introducing filter masks and won VOT 2017 challenge in real-time track. However, their method is plain RGB and our RGBD tracker differs from their work significantly in the following terms. Firstly, their method is an RGB tracker without occlusion handling mechanism. Secondly, we update correlation filter mask using depth cues instead of spatial 2D priors and color segmentation. As we show in Section IV-D, our approach is clearly superior which can be attributed to the power of depth cue.

An overview of our DM-DCF algorithm is given below while Section III-A provides an introduction to DCF based tracking, Section III-B reports how the depth based DCF masks are created, Section III-C discusses the optimization of correlation filters with spatial constraints and Section III-D introduces our occlusion handling mechanism.

Algorithm 1 Depth Masked DCF

Require: Current frame I^t; Occlusion state S^{t-1}; Foreground and background depth distributions P_{fg}^{t-1}, P_{bg}^{t-1}; Tracker response threshold τ; K last responses $\boldsymbol{r}$
  if S^{t-1} is $false$ **then** {** Tracker part **}
    Run DCF tracker ($\mathbf{h^{t-1}}$) on I^t
    Calculate maximum filter response r_{max}^t
    Run occlusion detection to obtain S^t
    Calculate depth mask $\mathbf{M^t}$ (Sec. III-B)
  else {** Detector part **}
    Run full frame detection and obtain r_{max}^t
    if $r_{max}^t > \tau * mean(\boldsymbol{r})$ **then**
      $S^t \leftarrow false$
    else
      $S^t \leftarrow true$
    end if
  end if
  if S^t is $false$ **then** {** Mask update part **}
    Update distributions P_{fg}^t and P_{bg}^t using $\mathbf{M^t}$
    Update $\mathbf{h^t}$ using $\mathbf{M^t}$
    Update tracker response history $\boldsymbol{r}^{mod(t,K)} \leftarrow r_{max}^t$
  end if
  Proceed to the next frame

A. Correlation Filters

The problem to be solved in correlation filter based tracking is to find a suitable filter h at discrete time point t and sample point i that provides desired output y_i for given input image x_i which includes the target object. Desired output, y_i, can be constructed by using a small, dense, 2D gaussian at the centre of a tracked object image [3]. Optimization of the filter can be formulated as a ridge regression problem:

$$\frac{1}{2}\sum_i ||y_i - h^T x_i||^2 + \frac{\lambda}{2}||h||^2 \tag{1}$$

where λ is the regularization parameter that is used to avoid overfitting to the current object appearance. A widely used technique is to assume circular repetitions of each input x_i as $x_i(\Delta\tau_j)$ where $\Delta\tau_j$ represents all circular shifts of x_i. This assumption leads to a fast filter optimization in the Fourier domain [11]:

$$\hat{h} = \frac{\sum_i \hat{x_i}^* \odot \hat{y_i}}{\sum \hat{x_i}^* \odot \hat{x_i} + \lambda} \tag{2}$$

where $\hat{h}$, $\hat{x}$ and $\hat{y}$ are the Fourier transforms of the correlation filter, input image and the desired output respectively. $\odot$ is the element-wise Hadamar product and * denotes the Hermitian conjugate. Computation in Fourier domain reduces the complexity from $\mathcal{O}(D^3 + ND^2)$ in the spatial domain to $\mathcal{O}(ND\log D)$ for the images of the size D pixels and N examples as it is reported in [9]. However, this also enforces a special form of (1):

$$\frac{1}{2}\sum_i\sum_j ||y_i(j) - h^T x_i(\Delta\tau_j)||^2 + \frac{\lambda}{2}||h||^2 \tag{3}$$

where j runs over the all D circular shifts of each input x_i. Henriques *et al.* [11] also extended the above to include kernel functions that can make tracker even more effective without any loss in computation speed.

Another important part of correlation filter based tracking is time averaging for online adaptation where previous appearances are retained in "tracker memory" [3]

$$\hat{h}^t = \psi\hat{h}^t + (1-\psi)\hat{h}^{t-1} \tag{4}$$

B. Depth Masking

The masking approach in our work to select active pixels (i.e. pixels that are used for DCF updates) for the DCF tracker was inspired by the work of Lukezic *et al.* [15] who constructed an RGB cue driven mask by forming a pixel graph where the foreground was segmented by graph cut approach using color histograms and spatial relationships. However, their method is deemed to fail in the cases where background and foreground are of similar color and it cannot detect occlusions as it can be observed in Fig. 2 and Fig. 3. On the other hand, in our method the depth cue turns out to be clearly better in mask generation and also provides an intuitive way to detect occlusion and switching from the tracking to detection mode which provides superior performance in long-term occlusions as in Fig. 2. In the case of foreground masking of a tracked object, the correlation filter is changed to a masked correlation filter, $h_M = M \odot h$, which replaces h in (1) or (3). The mask M has value 1 to mark the visible (active) region of a tracked object and value 0 to inactivate pixels in the background. Another advantage of masked correlation is the fact that the border effects in cyclic correlation can be removed if the mask is made larger than the current bounding box [9] (up to the size of the whole input frame).

We construct our mask using probabilistic representations P_{fg} and P_{bg} of foreground object and background (note that background in our case means scene elements both closer and

further away from the object) respectively. In its simplest form, the mask can be generated from foreground probability ratios

$$M(\boldsymbol{x}) = \begin{cases} 1, \text{if } \frac{P_{fg}}{P_{bg}} > \Omega \\ 0, \text{if } \frac{P_{fg}}{P_{bg}} \leq \Omega \end{cases} \tag{5}$$

However, we found another approach based on adaptive thresholding to work better since it avoids setting the threshold τ. We assign each mask pixel a probability ratio value $\log \frac{P_{fg}}{P_{bg}}$ which produces a "foreground probablity image" and the probability image is thresholded to form a binary foreground mask by the adaptive method of Otsu [18].

For the probability distribution estimation we tested both single Gaussian and Gaussian Mixture Models, but found single Gaussians to perform better and another additional benefit is their fast online update rules. Our distributions are $P_{fg} \propto \mathcal{N}(\mu_{fg}, \sigma^2_{fg})$ and $P_{bg} \propto \mathcal{N}(\mu_{bg}, \sigma^2_{bg})$ whose parameters are updated by the following rules:

$$\begin{aligned} \mu^{(t)} &= \mu^{(t)}\theta + (\mu^{(t-1)}(1-\theta)) \\ \sigma^{(t)} &= \sigma^{(t)}\gamma + (\sigma^{(t-1)}(1-\gamma)) \end{aligned} \tag{6}$$

where θ and γ are fixed update rates.

To construct the new distribution for the foreground, the depth values that are in the current mask are picked. In the first frame however, the ground truth bounding box provided by the dataset is used to create the initial distributions.

C. Filter Optimization

A mask generated by the procedure in Section III-B changes the target function (1) to find the optimal correlation filter h into

$$\frac{1}{2}\sum_i ||y_i - h^T M x_i||^2 + \frac{\lambda}{2}||h||^2 \tag{7}$$

and the circular function (3) into

$$\frac{1}{2}\sum_i \sum_j ||y_i(j) - Mh^T x_i(\Delta\tau_j)||^2 + \frac{\lambda}{2}||h||^2 \tag{8}$$

that can be written in the Fourier domain as [9]

$$E(h) = \frac{1}{2}\sum_i ||\hat{y}_i - \text{diag}(\hat{x}_i)^T \sqrt{D}\boldsymbol{F}M^T h||^2 + \frac{\lambda}{2}||h||^2 \tag{9}$$

where $M^T h$ are defined in the spatial domain and $D\boldsymbol{F}$ is the Fourier operator matrix $\boldsymbol{F}$ multiplied by the number of dimensions D in the signal. This solution is very inefficient in the Fourier domain ($\mathcal{O}(D^3 + ND^2)$) and therefore the primal solution in the spatial is faster. However, (9) is in the form where the Lagrangian multiplier $\hat{g}$ can be introduced and Alternating Direction Method of Multipliers (ADMM) adopted [4]:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\sum_i ||\hat{y}_i - \text{diag}(\hat{x}_i)^T \hat{g}||^2 + \frac{\lambda}{2}||h||^2 \\ &\text{subject to } \hat{g} = \sqrt{D}\boldsymbol{F}M^T h \end{aligned} \tag{10}$$

The augmented Lagrangian method uses the following unconstrained objective

$$\begin{aligned} &\text{minimize } \frac{1}{2}\sum_i ||\hat{y}_i - \text{diag}(\hat{x}_i)^T \hat{g}||^2 + \frac{\lambda}{2}||h||^2 \\ &+ \frac{\mu}{2}\left(\hat{g} - \sqrt{D}\boldsymbol{F}M^T h\right)^2 + \xi\left(\hat{g} - \sqrt{D}\boldsymbol{F}M^T h\right) \end{aligned} \tag{11}$$

where μ is the penalty term affecting the convergence and ξ is the Lagrange multiplier updated on each iteration. Optimization iteratively updates the estimates $\hat{g}^{t+1}$ and h^{t+1} and the multiplier using the rule

$$\xi^{t+1} = \xi^t - \mu\left(\hat{g}^{t+1} - \sqrt{D}\boldsymbol{F}M^T h^{t+1}\right) \tag{12}$$

D. Occlusion Detection

Detecting heavy occlusions ($\geq 90\%$) and consequently stopping model updates is a vital part of occlusion-aware tracking. This process allows DCF to avoid possible model pollution which eventually leads to drifting. In RGB-based DCF, the main source of information for occlusion handling is a correlation filter response at the maximum location r_{max} since a rapid decrease can be considered as an evidence of occlusion [1], [10]. To include tracker based occlusion detection, we calculate running mean of the maximum responses where the maximum of the current frame r^{curr}_{max} is added iteratively:

$$r^{(t+1)}_{max} = r^{(t)}_{max} + \frac{r^{curr}_{max} - r^{(t)}_{max}}{t} \tag{13}$$

The main drawback of the above tracker response based occlusion detection is the implicit assumption that occlusions occur faster than model appearance changes. This does not have to be true and might cause false occlusion detections.

To compensate the weakness of filter response based occlusion handling, we introduce a depth cue based occlusion detection which is simple and efficient. Intuitively, all pixels that pass through our probability based mask generation in Section III-B represent depth values where the target object appears. We can easily define the amount of occlusion to be allowed by enforcing a threshold for the visible pixels in M (10% in all our experiments). This depth based occlusion detection comes without any extra cost since the information is already available from the masking stage.

Our final occlusion detection combines both the *filter response based occlusion detection* and *depth based occlusion detection* where an occlusion is declared if both detectors are triggered, i.e. filter response falls below 65% of moving average and number of pixels supporting object depth falls as well below 10% of bounding box regions. If occlusion is detected, the filter update is stopped and the system switches into full image detection mode (occlusion recovery). Occlusion recovery model does not make any assumptions on object's reappearance probability and it is run as long as the target object is absent in the scene.

IV. Experiments

In this section, we present an extensive evaluation of the proposed method. Section IV-A provides implementation details, Section IV-B overviews the dataset and the metrics used for the evaluation, Section IV-C discusses the results and Section IV-D compares different variants of the proposed method in an ablation study.

A. Implementation Details

To make our results directly comparable to the state-of-the-art, we selected the same three RGB features in [15] (CSR-DCF): HOG [6], Color Names [22] and gray level pixel values. We also adopt the same parameter values as in the original CSR-DCF except DCF filter update rate (ψ) is set to 0.03. Update rates (θ, γ) for probability distributions P_{fg} and P_{bg} are set to 0.95 and 0.20 respectively. The parameters were kept fixed in our experiments that were run on non-optimized Matlab code with Intel Core i7 3.6GHz laptop and Ubuntu 16.04 OS. Our processing speed is calculated according to an average sequence where the number of occluded frames makes 25% of all frames.

B. Dataset and Evaluation Metrics

Princeton RGBD dataset [21] consists of 100 sequences from 11 categories and the authors provide ground truth only for five videos. Methods are evaluated by uploading them to a specific evaluation server. Results for other methods were taken from the online leaderboard table at the Princeton website with the exception of An *et al.* [1] who have not registered their method. Therefore, we took their numbers directly from the respective paper.

Bibi *et al.* [2] and Hannuna *et al.* [10] reported that 14% of Princeton RGBD dataset videos have synchronization errors between the RGB and depth frames. In addition, 8% of the sequences require bounding box re-alignment as pixel correspondences between RGB and depth frames were erroneous. In their experiments, Hannuna *et al.* and Bibi *et al.* used rectified versions of the dataset and therefore we found it fair to use their corrected sequences in our evaluations.

The evaluation uses the widely adopted *Intersection over Union* (IOU) metric proposed by the authors of the Princeton dataset similar to the one used in VOT RGB dataset [14].

C. Results

The results of our and the best performing other trackers for the Princeton RGBD dataset are given in Table I. As it can be seen below, our method performs on par with the top performing RGBD trackers (OAPF, RGBDOcc+OF and 3D-T) and is an order of magnitude faster. Out of the fast trackers (ours, DLST, DS-KCF and CSR-DCF) ours and DLST are the best with equal average rank, but our method is faster. In addition to that, our method wins in two categories: *Small* and *Passive*. These results indicate that our depth masked DCF is a suitable tracker for applications where balance between performance and speed is important.

The advantages of using the depth channel to complement 2D information are evident as our DM-DCF outperforms its RGB competitor, CSR-DCF, in almost all categories with a clear margin. The only category that CSR-DCF performs better is the "no occlusion" category where benefits of depth cue are understandably not necessary. Compared to the other DCF based methods DS-KCF-Shape [10] and DLST [1], DM-DCF performs considerably better in *Occlusion* category. This shows that our occlusion handling mechanism is more powerful as we use a maximum DCF response score history in conjunction with foreground segmentation using two separate probability distributions instead of a single frame response score and single distribution.

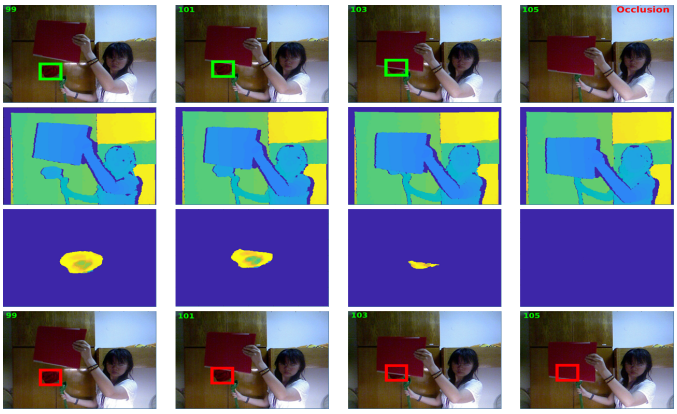

Fig. 3. Our method (top) can handle the challenging cases where the state-of-the-art DCF based RGB tracker (CSR-DCF [15]) fails due to the color similarity of a tracked object and back/foreground (bottom). Depth cue (second row) is used to generate a filter mask (third row) which is both used for filter generation and occlusion handling (note detected occlusion in the last frame when detection mode is switched on).

D. Ablation Study

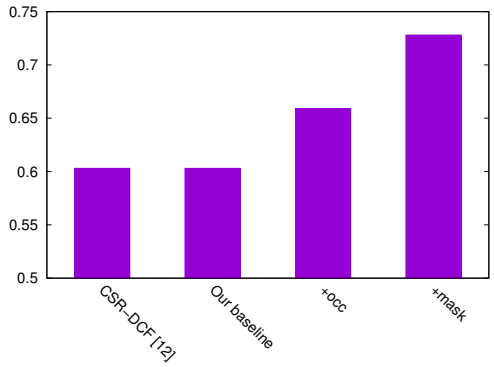

Fig. 4. Ablation study of the components of our methods.(Numbers represent the average accuracy over 11 categories in the dataset)

We conducted a set of ablation studies to support our design choices. Moreover, we also evaluated our algorithm on the original Princeton Dataset which has considerable amount of

TABLE I
COMPARISON OF THE BEST PERFORMING METHODS (ONLINE EVALUATION) FOR PRINCETON RGBD DATASET [21].

Alg.	Avg Rank	*Human*	*Animal*	*Rigid*	*Large*	*Small*	*Slow*	*Fast*	*Occl.*	*¬Occl.*	*Pass. Motion*	*Act. Motion*	*FPS*
3D-T [2]	2.81	**0.81 (1)**	0.64 (4)	0.73 (5)	**0.80 (1)**	0.71 (3)	0.75 (5)	**0.75 (1)**	**0.73 (1)**	0.78 (5)	0.79 (3)	0.73 (2)	N.A
RGBDOcc+OF [21]	3.27	0.74 (4)	0.63 (5)	**0.78 (1)**	0.78(3)	0.70 (4)	0.76 (2)	0.72 (3)	0.72 (2)	0.75 (6)	0.82 (2)	0.70 (4)	0.26
OAPF [16]	3.45	0.64 (6)	**0.85 (1)**	0.77 (3)	0.73 (5)	0.73 (2)	**0.85 (1)**	0.68 (6)	0.64 (6)	**0.85 (1)**	0.78 (4)	0.71 (3)	0.9
Our	3.63	0.76 (3)	0.58 (6)	0.77 (2)	0.72 (6)	**0.73 (1)**	0.75 (4)	0.72 (4)	0.69 (3)	0.78 (4)	**0.82 (1)**	0.69 (6)	**8.3**
DLST [1]	3.63	0.77 (2)	0.69 (3)	0.73 (6)	0.80 (2)	0.70 (6)	0.73 (6)	0.74 (2)	0.66 (4)	0.85 (2)	0.72 (6)	**0.75 (1)**	4.6
DS-KCF-Shape [10]	4.18	0.71 (5)	0.71 (2)	0.74 (4)	0.74 (4)	0.70 (5)	0.76 (3)	0.70 (5)	0.65 (5)	0.81 (3)	0.77 (5)	0.70 (5)	35.4
CSR-DCF [15]	10.55	0.53 (9)	0.56 (11)	0.68 (12)	0.55 (12)	0.62 (9)	0.66 (12)	0.56 (10)	0.45 (14)	0.79 (6)	0.67 (12)	0.56 (9)	13.6

registration and synchronization errors. We report the accuracy for the following variants:

- *CSR-DCF* State-of-the-art RGB tracker by Lukezic *et al.* [15]
- *DM-DCF–* Our method with the all proposed components switched off.
- *+occ* Depth based occlusion handling switched on.
- *+mask* Depth based masking added i.e. full DM-DCF.

As it can be seen in Fig. 4, adding depth based masking and occlusion handling improve the results almost $\%15$.

E. Summary

An important finding for the future work is that in general, different algorithms favor certain categories and motion types. As compared the results of all methods in Table I, most of the methods favor rigid motion over non-rigid motion (rigid vs. animal categories). This can be explained by the fact that the parameters are kept constant for all 95 test sequences and the adopted parameters favors rigid object tracking. Shape changes and adaptation speeds are different for non-rigid objects such as animals.

Another similar problem can be seen in occlusion vs. no occlusion. Again, improvement in the occlusion sequences means slight degradation in tracker performance in no occlusion cases. These observations suggest us to adopt adaptive parameters in our future work so that the tracker would adjust its parameters on the fly according to the target object.

V. CONCLUSION

In this paper, we proposed a Depth Masked Discriminative Correlation Filter (DM-DCF) RGBD tracker that uses the depth cue to detect occlusions (enable switching from the tracking to the detection mode) and to construct a spatial mask that improves DCF tracking. To this end, we are the first to use depth based segmentation masks inherently in DCF formulation extracting target regions for filter updates. Comparison and ablation studies on the publicly available Princeton RGBD Benchmark dataset verified that our trackers is on pair with the state-of-the-art while providing clearly better frame rate as compared to the top performers.

REFERENCES

[1] N. An, X.-G. Zhao, and Z.-G. Hou. Online rgb-d tracking via detection-learning-segmentation. In *ICPR*, 2016.

[2] A. Bibi, T. Zhang, and B. Ghanem. 3d part-based sparse tracker with automatic synchronization and registration. In *CVPR*, 2016.

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

[4] S. Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3, 2010.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25, pages 564–567, 2003.

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

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

[8] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In *ECCV*, 2016.

[9] H. Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR*, 2015.

[10] S. Hannuna, M. Camplani, J. Hall, M. Mirmehdi, D. Damen, T. Burghardt, A. Paiement, and L. Tao. Ds-kcf: a real-time tracker for rgb-d data. In *Journal of Real-Time Image Processing*, 2016.

[11] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-Speed Tracking with Kernelized Correlation Filters. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 37, pages 1–14, 2014.

[12] C. Hester and D. Casasent. Multivariant technique for multiclass pattern recognition. In *Applied Optics*, volume 19, pages 1758–1761, 1980.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 34, pages 1409–1422, 2011.

[14] M. Kristan, J. Matas, G. Nebehay, F. Porikli, and L. Cehovin. A Novel Performance Evaluation Methodology for Single-Target Trackers. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 38, pages 2137–2155, 2016.

[15] A. Lukezic, T. Vojir, L. Cehovin, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, 2017.

[16] K. Meshgi, S. I. Maeda, S. Oba, H. Skibbe, Y. Z. Li, and S. Ishii. An occlusion-aware particle filter tracker to handle complex and persistent occlusions. *Computer Vision and Image Understanding*, 150:81–94, 2016.

[17] H. Nam and B. Han. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In *CVPR*, 2016.

[18] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 1979.

[19] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental visual tracking. *International Journal of Computer Vision (IJCV)*, 77:125–141, 2008.

[20] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. volume 36, pages 1442–1468, 2014.

[21] S. Song and J. Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *ICCV*, 2013.

[22] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 18, pages 1512–1523, 2009.

[23] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang. Structural sparse tracking. In *CVPR*, 2015.

[24] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Gustav Hager, Alan Lukezic, Abdelrahman Eldesokey and Gustavo Fernandez The Visual Object Tracking VOT2017 Challenge Results. In *ICCV*, 2017.

PUBLICATION

II

How to Make an RGBD Tracker ?

U. Kart, J.-K. Kämäräinen and J. Matas

*European Conference on Computer Vision (ECCV) Workshops*2018

How to Make an RGBD Tracker ?

Uğur Kart[1][0000−0002−3728−2473] Joni-Kristian Kämäräinen[1][0000−0002−5801−4371]
Jiří Matas[2][0000−0003−0863−4844]

[1] Laboratory of Signal Processing
Tampere University of Technology, Tampere, Finland
{ugur.kart,joni.kamarainen}@tut.fi
[2] The Center for Machine Perception
Czech Technical University, Prague, Czech Republic
matas@cmp.felk.cvut.cz

Abstract. We propose a generic framework for converting an arbitrary short-term RGB tracker into an RGBD tracker. The proposed framework has two mild requirements – the short-term tracker provides a bounding box and its object model update can be stopped and resumed. The core of the framework is a depth augmented foreground segmentation which is formulated as an energy minimization problem solved by graph cuts. The proposed framework offers two levels of integration. The first requires that the RGB tracker can be stopped and resumed according to the decision on target visibility. The level-two integration requires that the tracker accept an external mask (foreground region) in the target update. We integrate in the proposed framework the Discriminative Correlation Filter (DCF), and three state-of-the-art trackers – Efficient Convolution Operators for Tracking (ECOhc, ECOgpu) and Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF). Comprehensive experiments on Princeton Tracking Benchmark (PTB) show that level-one integration provides significant improvements for all trackers: DCF average rank improves from 18th to 17th, ECOgpu from 16th to 10th, ECOhc from 15th to 5th and CSR-DCF from 19th to 14th. CSR-DCF with level-two integration achieves the top rank by a clear margin on PTB. Our framework is particularly powerful in occlusion scenarios where it provides 13.5% average improvement and 26% for the best tracker (CSR-DCF).

Keywords: Visual Object Tracking · RGBD Tracking

1 Introduction

Short-term visual object tracking has been an active research topic in computer vision due to its widespread application areas. In recent years, the community has witnessed rapid development and seen many successful trackers emerging thanks to standardized evaluation protocols and publicly available benchmarks and competitions [1–6]. In order to adapt to target appearance changes, short-term trackers update their tracking models over time. However, that makes them

prone to model corruption and drifting in case of persistent occlusions when the tracker adapts to the occluding object and starts to track it.

To avoid corruption, a tracker should be able to discriminate between the target object and the rest of the scene so that it can stop model updating if the target is occluded. However, this is a challenging task in the RGB space if there are occluders with similar visual appearance. To alleviate this issue, adding the depth cue to short-term trackers is intuitive; even if a tracked object is occluded by another object with similar appearance, the difference in their depth levels will be distinctive and will help to detect the occlusion. The availability of affordable depth sensors makes adoption of the depth cue even more attractive.

Since the depth channel lacks texture, depth itself may not provide useful information for visual tracking. On the other hand, RGB trackers perform competitively as long as no occlusions occur (see Table. 1). Therefore, our work aims at benefiting from the huge amount of effort that has been put on generic short-term RGB trackers and adopts depth as means for occlusion detection. As a novel solution, we propose a generic framework that can be used with any VOT compliant [6] short-term tracker to convert it into an RGBD tracker with depth-augmented occlusion detection. By applying the proposed framework through a clear interface and not changing the internal structure of a short-term tracker, a fast integration is ensured and the framework will benefit from ever improving short-term tracker performance in the future.

The proposed framework contains two main components: *short-term failure detection* and *recovery from occlusion*. Short-term failure detector continuously evaluates the target region to decide whether to allow the short-term tracker to update its model or switch to the recovery from occlusion mode. The framework also contains an optional, powerful third component which can be used with RGB trackers that accepts *foreground masks* that explicitly indicate occluded regions that do not belong to the target(*e.g.* CSR-DCF [7]).

The main contributions of this paper are:

- A generic framework to convert an arbitrary RGB short-term tracker into an RGBD tracker.
- Formulation of the framework's core component - *non-occluded foreground segmentation* - as an energy function of three terms, depth, color and spatial prior, which is optimized using graph cuts.
- RGBD versions of one baseline and three state-of-the-art short-term RGB trackers: DCF [8], ECO (ECOhc and ECOgpu variants) [9] and CSR-DCF [7]

The rest of the paper is organized as follows; Section 2 summarizes existing literature on generic, short-term tracking and RGBD trackers, Section 3 explains the proposed framework in detail, Section 4 provides the experiments and finally Section 5 concludes the paper.

2 Related Work

The aim of the provided generic framework is to convert any existing short-term RGB tracker into an RGBD tracker. We are motivated by the facts that the field

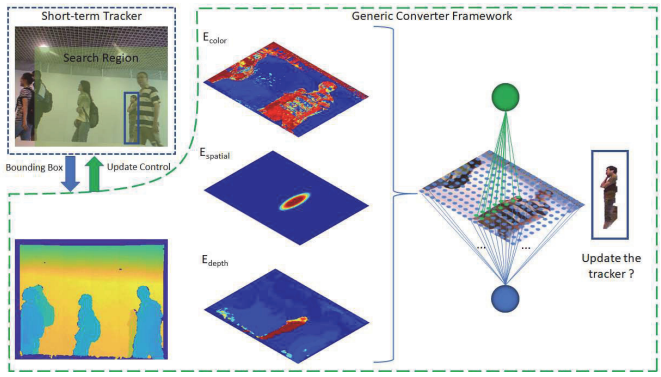

Fig. 1. Overview of the proposed framework. The short-term RGB tracker provides bounding box coordinates to the framework to be used for segmenting the visible target region with the help of depth. Using the ratio of the visible region to the bounding box, occlusions are detected hence, short-term tracker update is stopped and recovery mode is started. If the object is re-detected during recovery, the RGB tracker is resumed.

of short-term RGB tracking progresses on a steady basis and RGB provides a strong cue for tracking. On the other hand, we also believe that depth can be used as a complementary cue to instruct when a short-term tracker should be stopped and switched to recovery mode. In this sense, the proposed framework benefits from state-of-the-art short-term trackers which are briefly surveyed in addition to the recent RGBD trackers.

RGB Trackers – generic, short-term visual object tracking on RGB videos is a well-established research topic in computer vision and the main approaches can be grouped under two main categories. In *Generative Trackers*, a target model is stored and the goal is to find the best matching region in the next frame. A few descriptive examples for this category are Incremental Visual Tracking (IVT) [10], Structural Sparse Tracking (SST) [11] and kernel-based object tracking [12]. On the other hand, *Discriminative Trackers* continuously train a classifier using positive and negative samples that are acquired during the tracking process. Prominent examples of this category are Tracking-Learning-Detection (TLD) [13], Continuous Convolutional Operators Tracking (CCOT) [14], Multi-Domain Convolutional Neural Networks (MDNet) [15], Efficient Convolution Operators for Trackers (ECO) [9], and Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF) [7]. Due to their success in the last few years, discriminative trackers have been widely adopted in the recent works. For example, in the VOT 2017 challenge, 67% of the submissions were from this category [6]. However, training a classifier can be computationally expensive which has prompted the adoption of simple yet powerful methods for the training stage. Starting with the seminal work of Bolme *et al.* [16], Discrim-

inative Correlation Filter (DCF) based trackers have gained momentum due to their performance, fast model update (training) and mathematical elegance. Henriques *et al.* [17] proposed a method for efficient training of multiple samples that improves performance while providing very high FPS. To suppress the border artefacts resulting from circular correlation, Galoogahi *et al.* [8] posed the DCF learning as a more complex optimization problem which can still be efficiently solved with the help of the Augmented Lagrangian Method (ALM). Lukezic *et al.* [7] further improved their idea by introducing spatial reliability maps to extract unpolluted foreground masks. In VOT 2017, DCF based algorithms constitute almost 50% of the submitted trackers [6] with ECO [9] and CSR-DCF [7] being among the top performers while CSR-DCF C++ implementation won the best real-time tracker award.

RGBD Trackers – as compared to generic, short-term tracking on RGB, RGBD tracking is a relatively unexplored area. This can be partly attributed to the lack of datasets with groundtruths until recently. Song *et al.* [18] captured and annotated a dataset consisting of 100 videos with an online evaluation system and their benchmark is still the largest available. They also provided multiple baseline algorithms under two main categories; *depth as an additional cue* and *point cloud tracking*. Depth as an additional cue trackers treat depth as an extra channel to HOG features [19] whereas point cloud tracking methods use 3D point clouds for generating 3D bounding boxes. Among the ten proposed variations the one with RGBD HOG features and boosted by optical flow and occlusion detector achieved the best performance.

The seminal work of Song *et al.* inspired many followups. Meshgi *et al.* [20] proposed an occlusion-aware particle filter based tracker that can handle persistent occlusions in a probabilistic manner. Bibi *et al.* [21] also used a particle filter framework with sparse parts for appearance modeling. In their model, each particle is a sparse, linear combination of 3D cuboids which stays fixed during the tracking. Without occlusion, they first make a coarse estimation of the target location using 2D optical flow and then sample particles over the rotation R and translation T spaces. Occlusion is detected by counting the number of points in the 3D cuboid representation. Success of the DCF approach naturally caught the attention in the RGBD community as well. To the best of our knowledge, the first DCF based RGBD tracker was proposed by Camplani *et al.* [22]. They first cluster the depth histogram and then apply a single Gaussian distribution to model the tracked object in the depth space. To extract the foreground object, they assume that the cluster with the smallest mean is the object. The second method using DCF was proposed by An *et al.* [23] where Kernelized Correlation Filters (KCF) are used in conjunction with depth based segmentation for target localization. Heuristic approaches were adopted for detecting whether an object is in the occlusion state or not.

Recently, Kart *et al.* [24] proposed an algorithm for using the depth as a means to generate masks for DCF updates. Although this work is in a similar spirit, our method differs from theirs in multiple, fundamental aspects; first of all, the authors incorporate neither color nor spatial cues for the mask creation. This

results with the loss of very vital information sources. Especially in sequences where the target object and the occluding object have similar depth levels, it is very likely that the algorithm will not be able to discriminate in between even if they have different colors. Secondly, their foreground segmentation consists of a simple thresholding of depth probabilities which is an ad-hoc approach that requires careful fine tuning. Finally, the authors propose a brute force, full-frame grid search for recovering from the occlusion state whereas we propose to use the motion history of the target object to adaptively generate significantly smaller search areas to avoid redundant computational complexity.

3 RGBD Converter Framework

The proposed framework offers two levels of integration with the level-two being optional for trackers that can use a foreground mask in their model update. In the *level-one integration*, the framework continuously calculates the visibility state of a target object by casting the visibility problem as a pixel-wise foreground-background segmentation from multiple information sources: color, spatial proximity and depth. The segmentation result is the *foreground mask*. Without interfering with the internal structure of an RGB tracker, the framework uses the tracker output and bounding box to obtain a region of interest (ROI) for the segmentation step. If the ratio between the visible and occluded pixels is below a threshold, model updating of the RGB tracker is stopped and the framework goes into occlusion recovery mode. In the occlusion recovery mode, model update is stopped and the search region is continuously expanded around the last known location of the target. The search is performed by running the RGB tracker in a coarse-to-fine manner to find its maximum response r in the search region. The score is compared to the mean of last N valid responses (Section 3.4, Alg. 1). Once the target is detected, RGB tracker updating resumes. The *level-two integration* is available for trackers that use foreground masks in their model update.

For foreground segmentation, we adopt the energy minimization formulation in [25]:

$$E(f) = E_{smooth}(f) + E_{data}(f) \tag{1}$$

The goal is to find a pixel-wise labeling f (foreground/background) that minimizes the energy. E_{smooth} represents smoothness prior that penalizes neighboring pixels being labeled differently and E_{data} represents the observed data based energy. For E_{smooth}, we adopt the efficient computation of smoothed priors in [26] and E_{data} we formulate as

$$E_{data}(f) = E_{color}(f) + E_{spatial}(f) + E_{depth}(f) \tag{2}$$

where E_{color} measures the likelihood of observed pixel color given the target color model, E_{depth} models target region's depth and finally $E_{spatial}$ the spatial prior which is driven by the tracker location in the current frame. At the core of our approach are proper formulations of E_{color} (Section 3.1), E_{depth} (Section 3.2)

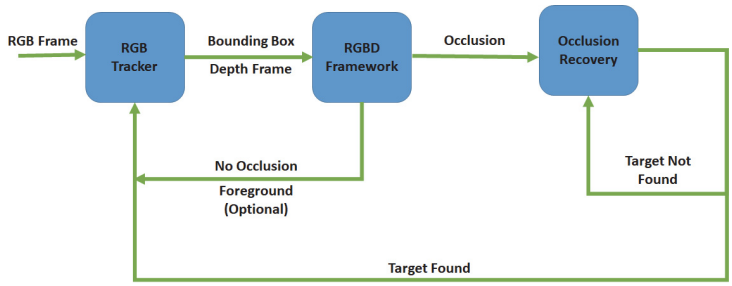

Fig. 2. The workflow diagram of the proposed framework. The framework uses bounding box coming from the RGB tracker and the depth frame to make a decision whether the target object is visible or not. In case it is visible, it allows the RGB tracker to update its model and continue tracking. If the target disappears, the framework runs the occlusion recovery module where the target object is searched using the last valid target model of the RGB tracker.

and $E_{spatial}$ (Section 3.3) so that the global optimum can be computed efficiently using the graph cuts algorithms [25, 27].

An example of the segmentation process is given in Fig. 3. As it can be seen, color based segmentation assigns both the target and the occluding object high confidence. However, the depth component is able to discriminate between the two while spatial component ensures high probability for the pixels that are close to the center of the tracking window.

3.1 Color-Based Target-Background Model E_{color}

In our formulation, E_{color} represents conditional dependencies between random variables (pixel fg/bg labels) for which we adopt a conditional random field formulation. The formulation uses the foreground/background probabilities as

$$E_{color} = \sum_{i \in \mathcal{V}} \psi_i(x_i) \tag{3}$$

where i is a graph vertex index (pixel) and x_i its corresponding label. ψ_i is encoded as a probability term

$$\begin{aligned} \psi_i(x_i = 0) &= -\log\left(p(x_i \notin fg)\right) \\ \psi_i(x_i = 1) &= -\log\left(p(x_i \in fg)\right) \end{aligned} \tag{4}$$

Since tracking is a temporal process, we need to add the frame number indicator to our notation $x_i \Rightarrow x_i^{(t)}$ where (t) is the current and $(t-1)$ the previous frame.

The probabilities $p(\cdot)$ can be efficiently computed using the color histograms of the foreground and background, h_f and h_b, respectively. It should be noted

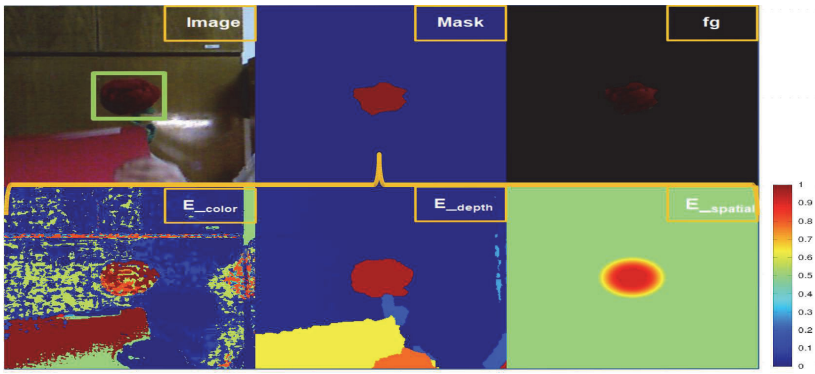

Fig. 3. Energy components in (2) and the segmentation output. The depth provides a strong cue even if the tracked and the occluding object have a very similar appearance.

that these histograms are updated after processing every frame for adapting to appearance changes. Therefore during processing frame t, the most recent color histograms are represented as h_f^{t-1} and h_b^{t-1}. Now, the color probability term is

$$p\left(x_i^t \in fg\right) = p\left(x_i^t = 1 \mid \text{hsv}(x_i^t), h_f^{(t-1)}, h_b^{(t-1)}\right) \quad . \tag{5}$$

where the $hsv(\cdot)$ function returns the HSV color space value of the pixel corresponding the label x_i in the current frame. The histograms are computed in 3D using $8 \times 8 \times 8 = 512$ uniformly distributed bins.

3.2 Depth-Based Target-Background Model E_{depth}

We model the depth induced energy E_{depth} similar to color using depth histograms $\hat{h}_f$ and $\hat{h}_b$

$$p\left(x_i^t \in fg\right) = p\left(x_i^t = 1 \mid \text{depth}(x_i^t), \hat{h}_f^{(t-1)}, \hat{h}_b^{(t-1)}\right) \tag{6}$$

where the depth probability is defined via the Bayesian rule (we use d to denote $\text{depth}(x_i^{(t)})$ for more compact representation)

$$p\left(x_i^{(t)} = 1 \mid d, \hat{h}_f^{(t-1)}, \hat{h}_b^{(t-1)}\right) = \frac{p\left(x_i^{(t)} = 1 \mid d, \hat{h}_f^{(t-1)}\right)}{p\left(x_i^{(t)} = 1 \mid d, \hat{h}_f^{(t-1)}\right) + p\left(x_i^{(t)} = 0 \mid d, \hat{h}_b^{(t-1)}\right)} \tag{7}$$

The above depth histograms are computationally efficient, but strongly biased against unseen depth levels. To be more precise, since probabilities for previously seen depth levels are high, the current frame (t) pixels with the same

depth levels are more likely to be assigned to the foreground and the model easily fails to introduce new depth levels. For tackling this problem, we add foreground and background distribution priors in the spirit of Bayesian estimation theory. For the foreground histogram estimation prior, we adopt the triangle function which has a maximum at the foreground depth mode (d denotes depth(x_i) for more compact notation and $||\cdot||$ is the length of the histogram)

$$\Psi_f(d) = tri(d) = \left(1 - \frac{|d - \text{mode}(\hat{h}_f^{(t-1)})|}{||\hat{h}_f^{(t-1)}||}\right) * \gamma \tag{8}$$

and for the background histogram estimation, we adopt the uniform distribution as a non-informative prior

$$\Psi_b(d) = \text{unif}(x_i) = \frac{1}{||\hat{h}_b^{(t-1)}||} * \theta \tag{9}$$

γ and θ are constants that control the prior gains. The choice of using a triangle distribution for foreground and uniform distribution for background stems from the following; in case of the foreground depth levels, it is expected that the newly seen depth levels will be similar to the current depth (e.g. a rotating object) and depth values in general are concentrated around the mode/mean. However, we cannot make any assumptions about the background and therefore we adopt the non-informative prior in (9).

To ensure continuous depth levels while not compromising from quick updates, we propose to apply a smoothening filter $g^t(d)$ to the observed histogram in the updating stage where $g^t(d)$ is a single Gaussian function centered at the histogram mode. By suppressing depth values that are highly unlikely to belong to the current observation, it provides a safety mechanism against wrong detections and drifting. Thus, the depth histogram updating process takes the following online update form:

$$\begin{aligned} \hat{h}_f^{(t)} &= \alpha \hat{h}_f^{(t-1)} + \left((1-\alpha)\hat{h}_f^{(t)}\right) \odot g^t(d) \\ \hat{h}_b^{(t)} &= \alpha \hat{h}_b^{(t-1)} + \left((1-\alpha)\hat{h}_b^{(t)}\right) \end{aligned} \tag{10}$$

3.3 Spatial Prior $E_{spatial}$

The third energy term in our model is a spatial prior that gives preference to foreground labels near the object center suggested by the short-term tracker;

$$p\left(x_i^t \in fg\right) = p\left(x_i^t = 1 \mid \boldsymbol{x}(x_i^t)\right) = k\left(\boldsymbol{x}(x_i^t); \sigma\right) \tag{11}$$

where $\boldsymbol{x}(\cdot)$ provides the spatial location (x,y) of the label x_i and $k(x;\sigma)$ is a clipped Epanechnikov kernel commonly used in kernel density estimation.

3.4 Occlusion Recovery

Given the energy terms E_{color}, E_{depth} and $E_{spatial}$, graph cut [25] provides labeling of each pixel in the tracker window by minimizing the energy. If the number of foreground pixels falls below a threshold τ, the tracker is stopped and recovery process started. To this end, we propose to use the trained RGB model M^t as an object detector since the depth information is no longer reliable, especially when the occlusion is persistent.

The proposed recovery strategy is based on three principles: (i) target object will be found again near the spatial location where it was previously seen, (ii) tracker response of a recovered object must be similar (proportional by Ω) to the previous tracked frames ($N = 30$ in our experiments), and (iii) each region must be expanded with a speed proportional to the object's average speed before the object was lost. By expanding the search region adaptively, computational redundancy of processing irrelevant spatial regions is avoided. Algorithm 1 summarizes the occlusion recovery process.

Algorithm 1 Occlusion Recovery

Require: Current frame I^t, response threshold constant Ω and target information before occlusion: $\{\boldsymbol{x}_i, \boldsymbol{bb}_i, r_i\}_{i=t-1,\ldots,t-10}$ (centroid, bounding box and response)
Initialization: $n = 0$ {# of frames in recovery mode}
Compute target speed $S = \max\left(5, \sum_{i=t-10}^{t-1} ||\boldsymbol{x}_i - \boldsymbol{x}_{i-1}||\right)$
while max response $r^n < \Omega * \text{mean}(r_i)$ **do**
  Expand $W^n = n * S + 2 * \boldsymbol{bb}(1)$
  Expand $H^n = n * S + 2 * \boldsymbol{bb}(2)$
  Extract patch $I^{W^n \times H^n} \subset I^t$ centered at $\boldsymbol{x}_{t-1}$
  $n = n + 1$
  Find the tracker maximum response r^n in $I^{W^n \times H^n}$
  Move to the next frame $t + 1$
end while
Reset depth histograms: $\hat{h}_f^{(t)}$ and $\hat{h}_b^{(t)}$
Resume tracking with the short-term RGB tracker

3.5 Target-Background Mask Extension for CSR-DCF

This section is related to the *level-two integration* explained in the beginning of Section 3 and as the example case we use the CSR-DCF tracker in [7]. Since the original idea of Discriminative Correlation Filter (DCF) for tracking [28, 16], many improvements have been proposed. An efficient solution in the Fourier domain was proposed by Henriques *et al.* [29] and their work was followed by an important extension by Galoogahi *et al.* [8] who relaxed the assumption of circular symmetrical filters. These extensions were adopted in CSR-DCF [7] which constructs a reliability mask that is used to mask out background regions during

tracker updates. Intuitively, the CSR mask can be replaced with the proposed foreground mask which is the output of graph cuts optimization (see Figure 3 for an example mask). In our experiments, it turns out that this significantly improves the performance of CSR-DCF since the proposed depth-based mask avoids model pollution more effectively. The level-two integration of our framework to CSR-DCF is simple: CSR mask is replaced with the mask produced by minimizing (1).

4 Experiments

In this section, we present the results for various trackers augmented with the proposed framework. Four generic, short-term trackers due to their proven success and efficiency are chosen; DCF [8], ECO [9] and CSR-DCF [7]. Since ECO has two variants, we applied the proposed framework to both ECO-gpu (deep features) and ECO-hc (hand crafted features).

4.1 Experimental Setup

Implementation Details – All experiments were run on a single laptop using a non-optimized Matlab code with Intel Core i7 3.6GHz and Ubuntu 16.04 OS. The parameters for the proposed algorithm were empirically set and kept constant during the experiments. Tracking parameters were as in the original works with the exception of DCF and CSR-DCF filter learning update rates were set to 0.03 and the number of bins for color histograms to 512. The rest of the parameters can be found in the publicly available code of our framework. [32]
Dataset – For validating the proposed framework we conducted experiments on the Princeton Tracking Benchmark (PTB) [18]. The dataset consists of 95 evaluation sequences and 5 validation sequences from 11 tracking categories, namely *human*, *animal*, *rigid*, *large*, *small*, *slow*, *fast*, *occlusion*, *no occlusion*, *passive motion* and *active motion*. The videos have been recorded with a standard Kinect 1.0 device and all frames annotated manually.
Evaluation Metrics – We use the metrics as they are provided by PTB [18].

However, the evaluation sequences do not contain publicly available ground truths except for the initial frame. To facilitate a fair comparison, Song *et al.* [18] also provide an online system where the resulting coordinates are uploaded for obtaining the final scores and ranking. The results of other methods in our paper were taken from the online system's website with the exception of DLST [23] who have not registered their methods. DLST scores were obtained from its paper. Bibi *et al.* [21] depth images are adopted in the experiments.

4.2 Comparison to State-of-the-Art

The results of the converted short-term trackers and the other top performing trackers on the PTB dataset are given in Table 1. Since the evaluation server did not allow multiple simultaneous submissions, we submitted each method

Table 1. Comparison of short-term RGB and RGBD tracking methods on the Princeton Tracking Benchmark (PTB) [18]. DCF [8] and three state-of-the-art trackers were used within the framework – ECOgpu [9], ECOhc [9] and CSR-DCF [7]; their level-one RGBD extensions are denoted DCF-rgbd, ECO-rgbd and CSR-DCF-rgbd, the level-two CSR-DCF integration where the original RGB-based mask is replaced by the proposed foreground mask (Section 3.5) is denoted CSR-DCF-rgbd++. (The table shows results for the Princeton Benchmark as of June 15, 2018)

	Avg Rank	**Tracking Category**										
Method		*Human*	*Animal*	*Rigid*	*Large*	*Small*	*Slow*	*Fast*	*Occ.*	*No-Occ.*	*Passive*	*Active*
⋆*CSR-DCF-rgbd++*	**3.64**	0.77(2)	0.65(5)	0.76(6)	0.75(4)	**0.73(1)**	0.80(3)	0.72(3)	0.70(3)	0.79(5)	0.79(5)	0.72(3)
OAPF [20]	5.27	0.64(14)	**0.85(1)**	0.77(4)	0.73(6)	0.73(2)	**0.85(1)**	0.68(8)	0.64(8)	**0.85(1)**	0.78(9)	0.71(4)
3D-T [21]	5.36	**0.81(1)**	0.64(7)	0.73(15)	**0.80(1)**	0.71(6)	0.75(8)	**0.75(1)**	**0.73(1)**	0.78(11)	0.79(6)	0.73(2)
RGBDOcc+OF [18]	5.55	0.74(5)	0.63(9)	0.78(2)	0.78(3)	0.70(7)	0.76(5)	0.72(4)	0.72(2)	0.75(17)	0.82(2)	0.70(5)
∘*ECOhc-rgbd*	6.18	0.70(7)	0.55(15)	**0.81(1)**	0.69(9)	0.72(4)	0.78(4)	0.68(7)	0.65(6)	0.79(6)	**0.83(1)**	0.66(8)
DSKCF-Shape [30]	6.64	0.71(6)	0.71(3)	0.74(11)	0.74(5)	0.70(8)	0.76(6)	0.70(6)	0.65(7)	0.81(4)	0.77(11)	0.70(6)
DLST [23]	6.73	0.77(3)	0.69(4)	0.73(16)	0.80(2)	0.70(9)	0.73(14)	0.74(2)	0.66(5)	0.85(2)	0.72(16)	**0.75(1)**
DM-DCF [24]	6.73	0.76(4)	0.58(12)	0.77(5)	0.72(8)	0.73(3)	0.75(10)	0.72(5)	0.69(4)	0.78(13)	0.82(3)	0.69(7)
DSKCF [22]	9.36	0.67(10)	0.61(10)	0.76(8)	0.69(10)	0.70(10)	0.75(9)	0.67(9)	0.63(9)	0.78(12)	0.79(7)	0.66(9)
⋄*ECOgpu-rgbd*	9.82	0.66(11)	0.58(11)	0.76(7)	0.65(14)	0.71(5)	0.81(2)	0.64(14)	0.62(10)	0.77(14)	0.78(8)	0.65(12)
DSKCF-CPP [22]	10.36	0.65(12)	0.64(8)	0.74(12)	0.66(13)	0.69(12)	0.76(7)	0.65(13)	0.60(12)	0.79(7)	0.80(4)	0.64(14)
RGBD+OF [18]	11.36	0.64(15)	0.65(6)	0.75(9)	0.72(7)	0.65(17)	0.73(15)	0.66(10)	0.60(13)	0.79(8)	0.74(15)	0.66(10)
hiob [31]	11.64	0.53(19)	0.72(2)	0.78(3)	0.61(16)	0.70(11)	0.72(16)	0.64(15)	0.53(16)	0.85(3)	0.77(12)	0.62(15)
⋆*CSR-DCF-rgbd*	11.91	0.68(9)	0.57(13)	0.74(10)	0.68(11)	0.68(14)	0.74(12)	0.65(12)	0.62(11)	0.75(16)	0.77(10)	0.64(13)
∘*ECOhc* [9]	12.18	0.69(8)	0.56(14)	0.72(17)	0.67(12)	0.68(13)	0.74(11)	0.65(11)	0.59(14)	0.78(9)	0.74(14)	0.65(11)
⋄*ECOgpu* [9]	15.36	0.58(16)	0.54(16)	0.73(13)	0.59(18)	0.65(15)	0.73(13)	0.58(17)	0.51(17)	0.78(10)	0.69(17)	0.60(17)
•*DCF-rgbd*	15.45	0.64(13)	0.54(17)	0.73(14)	0.65(15)	0.65(16)	0.71(17)	0.63(16)	0.59(15)	0.74(18)	0.76(13)	0.61(16)
•*DCF* [8]	18.09	0.56(17)	0.52(19)	0.66(18)	0.60(17)	0.59(19)	0.65(18)	0.57(18)	0.48(18)	0.74(19)	0.68(18)	0.56(18)
⋆*CSR-DCF* [7]	18.36	0.54(18)	0.53(18)	0.64(19)	0.56(19)	0.59(18)	0.61(19)	0.56(19)	0.44(19)	0.76(15)	0.64(19)	0.55(19)

separately and generated the leaderboard using the official protocol; methods were first ranked in each category and then the average rank was calculated.

The symbols •, ⋆, ⋄, and ∘ in Table 1 mark the trackers that our framework applied to. As it can be observed, the proposed method clearly has a big impact on overall rankings for all three trackers. Especially in sequences with occlusions, this impact becomes more visible. CSR-DCF improves 8 ranks, ECOgpu ranking sees 7 ranks improvement and ECOhc rank improves by 8. In terms of accuracy, the improvement is as strong as in rankings. When the proposed framework (without foreground masked updates) is applied to CSR-DCF, its performance in occlusion sequences increases 18% while ECOhc grows by 6% and ECOgpu 11%. The level-two integration further boosts occlusion sequences accuracy for CSR-DCF to a total of 26%.

Unlike other top performing methods, CSR-DCF-rgbd++ also maintains a well-balanced performance over all the categories by staying among the top in every one. This suggests that it does not overfit to specific categories but it provides similar performance for different scenarios which makes it a very suitable candidate for real-life applications.

Fig. 4 shows that the proposed framework adds to tracker's occlusion resilience to both short-term and long-term occlusions. For example, ECOhc-rgbd was able to detect the occlusion and it also re-detected the target object when it reappeared in the scene instead of drifting due to model pollution. As a good

Fig. 4. Short-term and long-term occlusion examples comparing the original methods(red) and their RGBD versions(green). Top row – DCF, second row – ECOgpu, third row – ECOhc, bottom row – CSR-DCF.

example of long-term recovery example, CSR-DCF-rgbd++ was able to recover even after 35 frames of occlusion state since it avoided model corruption and expanded the search region gradually.

The reason why CSR-DCF-rgbd++ performs better than the other RGBD methods can be possibly explained by its masked DCF update mechanism which uses the foreground provided by the framework. In the discriminative tracking paradigm, the tracker's target model is updated over the time for coping with the visual changes. However, when a rectangular bounding box is used for this purpose, it is likely to include background and occluding object's pixels as well. This results with learning of irrelevant information that may cause drifting. Whereas in the masked update approach, the updates are done only using the pixels that are confidently belong to the target object. Thus, the target model stays uncorrupted which results with better performance.

5 Conclusions

A generic framework was proposed for converting existing short-term RGB trackers into RGBD trackers. The framework is easy to adopt as it only requires control of model updating (stop/resume) and a tracking bounding box which are both provided in any tracker that is VOT compliant [6]. At the core of the framework is a foreground model which uses depth, color and spatial cues to efficiently detect occluded regions which are utilized at two-levels: occlusion detection and optionally, masked tracker updates. In all experiments, existing RGB trackers improved their ranks in the publicly available Princeton tracking benchmark [18]. CSR-DCF tracker which allows level-two integration of the proposed foreground model achieved state-of-the-art accuracy and was ranked the best on the day of submission. The full source code of the framework is publicly available. [32]

Acknowledgements

Uğur Kart was supported by two projects: Business Finland Project "360 Video Intelligence - For Research Benefit" with Nokia, Lynx, JJ-Net, BigHill, Leonidas and Business Finland - FiDiPro Project "Pocket - Sized Big Visual Data". Jiří Matas was supported by the OP VVV MEYS project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

References

1. Wu, Y., Lim, J., Yang, M.H.: Online Object Tracking: A Benchmark. In: CVPR. (2013)
2. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., et al.: The Visual Object Tracking VOT2013 Challenge Results. In: CVPR Workshops. (2013)
3. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Cehovin, L., Nebehay, G., Vojir, T., et al.: The Visual Object Tracking VOT2014 Challenge Results. In: ECCV Workshops. (2014)
4. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., et al.: The Visual Object Tracking VOT2015 challenge results. In: ICCV Workshops. (2015)
5. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin, L., Vojir, T., et al.: The Visual Object Tracking VOT2016 Challenge Results. In: ECCV Workshops. (2016)
6. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., et al.: The Visual Object Tracking VOT2017 Challenge Results. In: ICCV Workshops (2017)
7. Lukezic, A., Vojir, T., Cehovin, L., Matas, J., Kristan, M.: Discriminative Correlation Filter with Channel and Spatial Reliability. In: CVPR. (2017)
8. Galoogahi, H., Sim, T., Lucey, S.: Correlation Filters with Limited Boundaries. In: CVPR. (2015)
9. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: ECO: Efficient Convolution Operators for Tracking. In: CVPR. (2017)
10. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental Visual Tracking. International Journal of Computer Vision (IJCV) **77** (2008) 125–141
11. Zhang, T., Liu, S., Xu, C., Yan, S., Ghanem, B., Ahuja, N., Yang, M.H.: Structural Sparse Tracking. In: CVPR. (2015)
12. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-Based Object Tracking. IEEE PAMI **25** (2003) 564–567
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-Detection. In: IEEE PAMI. Volume 34. (2011) 1409–1422
14. Danelljan, M., Robinson, A., Shahbaz Khan, F., Felsberg, M.: Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In: ECCV. (2016)
15. Nam, H., Han, B.: Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In: CVPR. (2016)
16. Bolme, D.S., Beveridge, J., Draper, B.A., Lui, Y.M.: Visual Object Tracking using Adaptive Correlation Filters. In: CVPR. (2010)
17. Henriques, J., Caseiro, R., Martins, P., Batista, J.: High-Speed Tracking with Kernelized Correlation Filters. In: IEEE PAMI. Volume 37. (2014) 1–14
18. Song, S., Xiao, J.: Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. In: ICCV. (2013)
19. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR. (2005)
20. Meshgi, K., Maeda, S.I., Oba, S., Skibbe, H., Li, Y.Z., Ishii, S.: An Occlusion-Aware Particle Filter Tracker to Handle Complex and Persistent Occlusions. CVIU **150** (2016) 81–94
21. Bibi, A., Zhang, T., Ghanem, B.: 3D Part-Based Sparse Tracker with Automatic Synchronization and Registration. In: CVPR. (2016)
22. Camplani, M., Hannuna, S., Mirmehdi, M., Damen, D., Paiement, A., Tao, L., Burghardt, T.: Real-time RGB-D Tracking with Depth Scaling Kernelised Correlation Filters and Occlusion Handling. In: BMVC. (2015)

23. An, N., Zhao, X.G., Hou, Z.G.: Online RGB-D Tracking via Detection-Learning-Segmentation. In: ICPR. (2016)
24. Kart, U., Kämäräinen, J.K., Matas, J., Fan, L., Cricri, F.: Depth Masked Discriminative Correlation Filter. In: ICPR. (2018)
25. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE PAMI **23**(11) (2001)
26. Diplaros, A., Vlassis, N., Gevers, T.: A spatially constrained generative model and an EM algorithm for image segmentation. IEEE Trans. on Neural Networks **18**(3) (2007)
27. Rother, C., Kolmogorov, V., Blake, A.: GrabCut interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. (2004)
28. Hester, C., Casasent, D.: Multivariant technique for multiclass pattern recognition. In: Applied Optics. Volume 19. (1980) 1758–1761
29. Henriques, J., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV. (2012)
30. Hannuna, S., Camplani, M., Hall, J., Mirmehdi, M., Damen, D., Burghardt, T., Paiement, A., Tao, L.: DS-KCF: A Real-Time Tracker for RGB-D Data. In: Journal of Real-Time Image Processing. (2016)
31. Springstübe, P., Heinrich, S., Wermter, S.: Continuous Convolutional Object Tracking. In: ESANN. (2018)
32. https://github.com/ugurkart/rgbdconverter

PUBLICATION III

Object Tracking by Reconstruction with View-Specific Discriminative Correlation Filters

U.Kart, A. Lukežič, M. Kristan, J.-K. Kämäräinen and J. Matas

*IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*2019

Object Tracking by Reconstruction with View-Specific Discriminative Correlation Filters

Uğur Kart*, Alan Lukežič†, Matej Kristan†, Joni-Kristian Kämäräinen*,Jiří Matas‡
*Laboratory of Signal Processing, Tampere University, Finland
† Faculty of Computer and Information Science, University of Ljubljana, Slovenia
‡ Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
{ugur.kart,joni.kamarainen}@tuni.fi
{alan.lukezic,matej.kristan}@fri.uni-lj.si
matas@cmp.felk.cvut.cz

Abstract

Standard RGB-D trackers treat the target as a 2D structure, which makes modelling appearance changes related even to out-of-plane rotation challenging. This limitation is addressed by the proposed long-term RGB-D tracker called OTR – Object Tracking by Reconstruction. OTR performs online 3D target reconstruction to facilitate robust learning of a set of view-specific discriminative correlation filters (DCFs). The 3D reconstruction supports two performance-enhancing features: (i) generation of an accurate spatial support for constrained DCF learning from its 2D projection and (ii) point-cloud based estimation of 3D pose change for selection and storage of view-specific DCFs which robustly localize the target after out-of-view rotation or heavy occlusion. Extensive evaluation on the Princeton RGB-D tracking and STC Benchmarks shows OTR outperforms the state-of-the-art by a large margin.

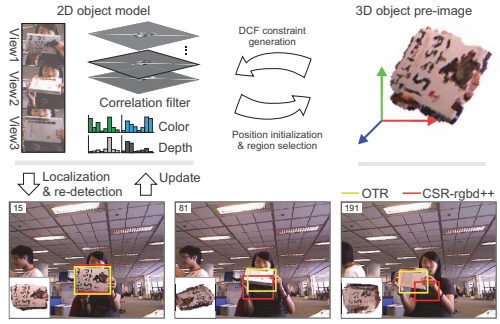

Figure 1. The OTR – Object Tracking by Reconstruction – object model consists of a set of 2D view-specific DCFs and of an approximate 3D object reconstruction. The OTR thus copes well with out-of-view rotation with a significant aspect change, while a state-of-the-art tracker CSR-rgbd++ [19] drifts and fails.

1. Introduction

Visual object tracking (VOT) is one of the core problems in computer vision; it has many applications [18, 8]. The field has progressed rapidly, fueled by the availability of large and diverse datasets [39, 34] and the annual VOT challenge [22, 23]. Until recently, tracking research has focused on RGB videos, largely neglecting RGB-D (rgb+depth) tracking as obtaining a reliable depth map at video frame rates has not been possible without expensive hardware. In the last few years, depth sensors have become widely accessible, which has lead to a significant increase of RGB-D tracking related work [6, 1, 26]. Depth provides important cues for tracking since it simplifies reasoning about occlusions and facilitates depth-based object segmentation. Progress in RGB-D tracking has been further boosted by the emergence of standard datasets and evaluation protocols [35, 40].

In RGB-D tracking, direct extensions of RGB methods by adding the D-channel as an additional input dimension have achieved considerable success. In particular, discriminative correlation filter (DCF) based methods have shown excellent performance on the Princeton RGB-D tracking benchmark [35], confirming the reputation gained on RGB benchmarks [22, 23, 19, 20, 6, 1]. Furthermore, DCFs are efficient in both learning of the visual target appearance model and in target localization, which are both implemented by FFT, running in near real time on a standard CPU.

A major limitation of the standard RGB and RGB-D trackers, regardless of the actual method (e.g. DCF [4], Siamese deep nets [2], Mean shift [9], Lucas Kanade [27]),

is that they treat the tracked 3D object as a 2D structure. Thus even a simple rotation of a rigid 3D object is interpreted as potentially significant appearance change in 2D that is conceptually indistinguishable from partial occlusion, tracker drift, blurring and ambient light changes.

Consider a narrow object, e.g., a book, with its front cover facing the camera, that rotates sideways and ends with its back side facing the camera (Figure 1). From the perspective of a standard RGB tracker, the object has deformed and the appearance has completely changed. Since most of the standard trackers cannot detect (do not model) aspect changes, the target bounding box and the appearance model contain mostly pixels belonging to the background when the narrow side of the book is facing the camera. Furthermore, the model update is carried out by implicit or explicit temporal averaging of the tracked views. Consequently, the appearance observed in the earlier frames is lost after a certain time period, limiting re-detection capability in situation when the target is completely occluded, but later re-appears, since its appearance no longer matches the last observed view. The above-mentioned problems are almost trivial to solve if a 3D model with attached photometric information is available for the tracked object.

We exploit the opportunity of using the depth component in RGB-D signal to build a simple, yet powerful 3D object representation based on the surface splat model, i.e., the object surface is approximated by a collection of 3D points with color, radius and the normal – *surfels*. This model has been proven very powerful in the context of SLAM [33]. The 3D model is aligned and updated to the current 2D target appearance during tracking by an ICP-based matching mechanism [33] – thus a *pre-image* of the 2D target projection is maintained during tracking. The 3D object pre-image significantly simplifies detection and handling of (self-)occlusion, out-of-plane rotation (view changes) and aspect changes.

The ICP-based 3D pre-image construction [33] requires accurate identification of the object pixels in the current frame prior to matching, and it copes with only small motions due to a limited convergence range. A method from a high-performance RGB-D DCF tracker [19] is thus used to robustly estimate potentially large motions and to identify object pixels for the pre-image construction. The DCF learning is improved by generating appearance constraints from the pre-image. Object appearance changes resulting from out-of-view rotation are detected by observing the pre-image 3D motion and a set of view-specific DCFs is generated. These 2D models are used during tracking for improved localization accuracy as well as for target re-detection using the recent efficient formulation of the DCF-based detectors [30]. The resulting tracker thus exhibits a long-term capability, even if the target re-appears in a pose different from the one observed before the occlusion.

Contributions The main contribution of the paper is a new long-term RGB-D tracker, called OTR – Object Tracking by Reconstruction that constructs a 3D model with view-specific DCFs attached. The DCF-coupled estimation of the object pre-image and its use in DCF model learning for robust localization has not been proposed before. The OTR tracker achieves the *state-of-the-art*, outperforming prior trackers by a large margin on two standard RGB-D tracking benchmarks. An ablation study confirms the importance of view-specific DCF appearance learning that is tightly connected to the 3D reconstruction. We will make the reference implementation of OTR available at https://github.com/ugurkart.

2. Related Work

RGB Tracking Of the many approaches proposed in the literature, DCF-based methods have demonstrated excellent performance – efficiency trade-off in recent tracking challenges [24, 22, 23]. Initially proposed by Bolme *et al*. [4], DCF-based tracking captured the attention of the vision community due to its simplicity and mathematical elegance. Improvements of the original method include multi-channel formulation of correlation filters [12, 15], filter learning using kernels exploiting properties of circular correlation [17] and scale estimation with multiple one-dimensional filters [11]. Following these developments, Galoogahi *et al*. [14] tackled the boundary problems that stem from the nature of circular correlation by proposing a filter learning method where a filter with size smaller than the training example is adopted. Lukezic *et al*. [28] further improved this idea by formulating the filter learning process using a graph cut based segmentation mask as a constraint.

RGB-D Tracking The most extensive RGB-D object tracking benchmark has been proposed by Song *et al*. [35] (Princeton Tracking Benchmark). The benchmark includes a dataset, evaluation protocol and a set of baseline RGB-D trackers. Several RGB-D trackers have been proposed since. Meshgi *et al*. [31] used an occlusion-aware particle filter framework. A similar approach was proposed by Bibi *et al*. [3] but using optical flow to improve localization accuracy. As an early adopter of DCF based RGB-D trackers, Hannuna *et al*. [16] used depth as a clue to detect occlusions while tracking is achieved by KCF [17]. An *et al*. [1] performed a depth based segmentation along with a KCF tracker. Kart *et al*. [20] proposed a purely depth based segmentation to train a constrained DCF similarly to CSR-DCF [28] and later extended their work to include color in segmentation [19]. Liu *et al*. [26] proposed a context-aware 3-D mean-shift tracker with occlusion handling. At the time of writing this paper [26] is ranked first at Princeton Tracking Benchmark. Xiao *et al*. [40] recently proposed a new RGB-D tracking dataset (STC) and an RGB-D tracker by

adopting an adaptive range-invariant target model.

3D Tracking Klein *et al*. [21] proposed a camera pose tracking algorithm for small workspaces which works on low-power devices. The approach is based on tracking keypoints across the RGB frames and bundle adjustment for joint estimation of the 3D map and camera pose. Newcombe *et al*. [32] proposed an iterative closest point (ICP) based algorithm for depth sequences for dense mapping of indoor scenes. In a similar fashion, Wheelan *et al*. [38] used surfel-based maps and jointly optimized color and geometric costs in a dense simultaneous localization and mapping (SLAM) framework. All three methods are limited to static scenes and are inappropriate for object tracking. This limitation was addressed by Rünz *et al*. [33], who extended [38] by adding the capability of segmenting the scene into multiple objects. They use a motion consistency and semantic information to separate the object from the background. This limits the method to large, slow moving objects.

Lebeda *et al*. [25] combined structure from motion, SLAM and 2D tracking to cope with 3D object rotation. Their approach reconstructs the target by tracking keypoints and line features, however, it cannot cope with poorly-textured targets and low-resolution images.

3. Object tracking by 3D reconstruction

In OTR, object appearance is modeled at two levels of abstraction which enables per-frame target localization and re-detection in the case of tracking failure. The appearance level used for localizing the target in the image is modelled by a a set of view-specific discriminative correlation filters, i.e., a DCF $\mathbf{h}_t$ that models the current object appearance, and a set of snapshots $\{\mathbf{h}^{(s)}\}_{s=1}^{S}$ modelling the object from previously observed views. In addition to the filters, the object color and depth statistics are modelled by separate color and depth histograms for the foreground and the background.

The second level of object abstraction is a model of the object pre-image $\Theta_t = \{\mathbf{P}_t, \mathbf{R}_t, \mathbf{T}_t\}$, where $\mathbf{P}_t$ is the surfel-based object 3D model specified in the object-centered coordinate system and $\{\mathbf{R}_t, \mathbf{T}_t\}$ are the rotation and the translation of the 3D model into the current object position.

The two models interact during tracking for improved DCF training and 3D pose change detection (e.g., rotations). We describe the DCF framework used by the OTR tracker in Section 3.1, the multi-view DCFs with the pre-image model is detailed in Section 3.2, Section 3.3 details target loss recovery and Section 3.4 summarizes the full per-frame tracking iteration.

3.1. Constrained DCF

The core DCF tracker in the OTR framework is the recently proposed constrained discriminative correlation filter CSR-DCF [29], which is briefly outlined here. Given a search region of size $W \times H$ a set of N_d feature channels $\mathbf{f} = \{\mathbf{f}_d\}_{d=1}^{N_d}$, where $\mathbf{f}_d \in \mathcal{R}^{W \times H}$, are extracted. A set of N_d correlation filters $\mathbf{h} = \{\mathbf{h}_d\}_{d=1}^{N_d}$, where $\mathbf{h}_d \in \mathcal{R}^{W \times H}$, are correlated with the extracted features and the object position is estimated as the location of the maximum of the weighted correlation responses

$$\mathbf{r} = \sum\nolimits_{d=1}^{N_d} w_d (\mathbf{f}_d \star \mathbf{h}_d), \tag{1}$$

where $\star$ represents circular correlation, which is efficiently implemented by a Fast Fourier Transform with $\{w_d\}_{d=1}^{N_d}$ being the channel weights. The target scale can be efficiently estimated by another correlation filter trained over the scale-space [11].

Filter learning is formulated in CSR-DCF as a constrained optimization that minimizes a regression loss

$$\varepsilon(\mathbf{h}) = \sum_{d=1}^{N_c} \|\mathbf{f}_d \star \mathbf{h}_d - \mathbf{g}\|^2 + \lambda \|\mathbf{h}_d\|^2 \; ; \; \mathbf{h}_d \equiv \mathbf{m} \odot \mathbf{h}_d, \tag{2}$$

where $\mathbf{g}$ is a desired output and $\mathbf{m}$ is a binary mask $\mathbf{m} \in \{0,1\}^{W \times H}$ that approximately separates the target from the background. The mask thus acts as a constraint on the filter support, which allows learning a filter from a larger training region as well as coping with targets that are poorly approximated by an axis-aligned bounding box. CSR-DCF applies a color histogram-based segmentation for mask generation, which is not robust to visually similar backgrounds and illumination change. We propose generating the mask from the RGB-D input and the estimated pre-image in Section 3.2.1.

Minimization of (2) is achieved by an efficient ADMM scheme [5]. Since the support of the learned filter is constrained to be smaller than the learning region, the maximum response on the training region reflects the reliability of the learned filter [28]. These values are used as per-channel weights w_d in (1) for improved target localization (we refer the reader to [29] for more details).

3.2. A multi-view object model

At each frame, the current filter $\mathbf{h}_t$ is correlated within a search region centered on the target position predicted from the previous frame following (1). To improve localization during target 3D motion, we introduce a "memory" which is implemented by storing captured snapshots $\{\mathbf{h}^{(s)}\}_{s=1}^{S}$ from different 3D view-points (i.e., a set of view-specific DCFs). At every N_{R}-th frame, all view-specific DCFs are evaluated, and the location of the maximum of the correlation response is used as the new target hypothesis $\mathbf{x}_t$. If the maximum correlation occurs in the set of snapshots, the current

filter is replaced by the corresponding snapshot filter. Target presence is determined at this location by the test described in Section 3.3.1. In the case the test determines target is lost, the tracker enters a re-detection stage described in Section 3.3.

If the target is determined to be present, the current filter $\mathbf{h}_t$ is updated by a weighted running average

$$\mathbf{h}_{t+1} = (1-\eta)\mathbf{h}_t + \eta\tilde{\mathbf{h}}_t, \tag{3}$$

where $\tilde{\mathbf{h}}_t$ is a new filter estimated by the constrained filter learning in Section 3.1 at the estimated position $\mathbf{x}_t$ and η is the update factor.

In addition to updating the current filter, the object color and depth histograms are updated as in [19], the object pre-image is updated as described in Section 3.2.2 and the set of view-specific DCFs $\{\mathbf{h}^{(s)}\}_{s=1}^{S}$ is updated following Section 3.2.3.

3.2.1 Object pre-image-based filter constraint

The binary mask $\mathbf{m}$ used in the constrained learning in (2) is computed at the current target position at filter learning stage. In the absence of other inputs, the mask is estimated by a recent segmentation approach from [19]. This approach uses an MRF segmentation model from CSR-DCF [29] within the filter learning region and estimates per-pixel unary potentials by color and depth (foreground/background) histograms backprojection in the RGB-D image.

However, the pre-image Θ_t can be used to better outline the object in the filter training region, leading to a more accurately learned filter. Thus, at DCF training stage, the pre-image is generated by fitting the object 3D model $\mathbf{P}_t$ onto the current object appearance (Section 3.2.2). If the fit is successful, the segmentation mask used in filter learning (2) is replaced by a new mask generated as follows. The 3D model $\mathbf{P}_t$ is projected into the 2D filter training region. Pixels in the region corresponding to the visible 3D points are set to one, while others to zero, thus forming a binary object occupancy map. The map is dilated to remove holes in the object mask and only the largest connected component is retained, while others are set to zero to reduce the effect of potential reconstruction errors in the 3D model. An example of the 2D mask construction from the 3D object pre-image is demonstrated in Figure 2.

3.2.2 Object pre-image update

The object pre-image Θ_t is updated from the object position estimated by the multi-view DCF (Section 3.2). Pixels corresponding to the target are identified by the color+depth segmentation mask from Section 3.2.1. The patch is extracted from the RGB-D image and used to update the object 3D model $\mathbf{P}_t$. The 3D model $\mathbf{P}_t$ is first translated to

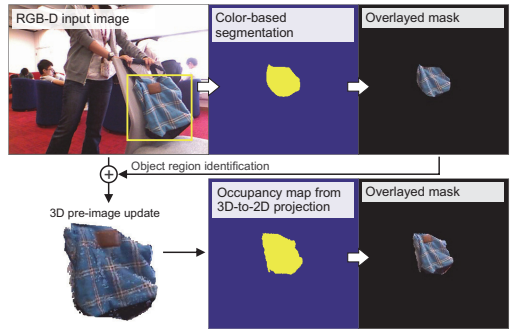

Figure 2. A 2D DCF localizes the target (top-left), the target color+depth pixels are approximately segmented (top-right) and used to update the 3D pre-image (bottom-left). The pre-image is projected to 2D generating an occupancy map (bottom-right). The resulting mask better delineates the object, which improves the constrained DCF learning.

the 3D position determined by the target location from the multi-view DCF. The ICP-based fusion from [33], that uses color and depth, is then applied to fine align the 3D model with the patch and update it by adding and merging the corresponding surfels (for details we refer to [33]). The updated model is only retained if the ICP alignment error is reasonably low (i.e., below a threshold τ_{ICP}), otherwise the update is discarded.

3.2.3 A multi-view DCF update

Continuous updates may lead to gradual drift and failure whenever the target object undergoes a significant appearance change. Recovery from such situations essentially depends on the diversity of the target views captured by the snapshots $\{\mathbf{h}^{(s)}\}_{s=1}^{S}$ and their quality (e.g., snapshots should not be contaminated by the background). The following conservative update mechanism that maximizes snapshot diversity and minimizes contamination is applied.

The current filter is considered for addition to the snapshots only if the target passed the presence test (Section 3.3.1) and the object pre-image Θ_t is successfully updated (Section 3.2.2). Passing these two tests, the target is considered visible with the pre-image accurately fitted. A filter is added if the object view has changed substantially and results in a new appearance (viewpoint). The change is measured by a difference between the reference aspect ρ_0 (i.e., a bounding box width-to-height ratio) and the aspect ρ_t obtained from the current 2D projection of the object pre-image. Whenever this difference exceeds a threshold, i.e., $\|\rho_0 - \rho_t\| > \tau_\rho$, a new snapshot is created and the current ratio becomes a new reference, i.e., $\rho_0 \leftarrow \rho_t$. In our preliminary experiments, we tested using Euler angles of the estimated rotation matrix $\mathbf{R}$, but this was found sensitive to

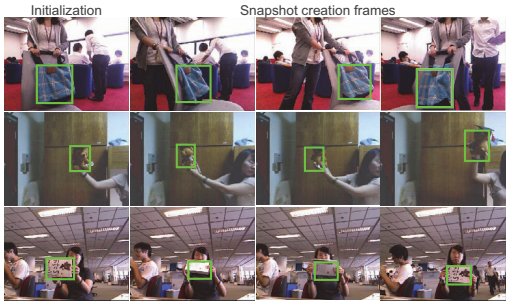

Figure 3. Examples of view-specific DCFs creation. The tracker was initialized on the images in the left-most column, while the remaining images represent frames in which a new view was detected and stored in the set of view-specific DCFs.

ICP estimation errors and therefore aspect ratio test proved to be more robust. Examples of images used to create separate DCF views are shown in Figure 3.

3.3. A multi-view DCF target detection

Target presence is determined at each frame using the test described in Section 3.3.1. Whenever the target is lost, the following re-detection mechanism is activated. At each frame all filters in the snapshot set $\{\mathbf{h}^{(s)}\}_{s=1}^{S}$ are correlated with features extracted from a region centered at the last confident target position. To encode a motion model, the search region size is gradually increased in subsequent frames by a factor $\alpha_s^{\Delta t}$, where $\alpha_s > 1$ is a fixed scale factor and Δt is the number of frames since the last confident target position estimation. The correlation is efficiently calculated by padding the snapshots with zeros to the current search region size and applying FFT [30].

Since the target may change the size, a two-stage approach for re-detection is applied. First, the hypothesized target position is estimated as the location of the maximum correlation response and the filter $\mathbf{h}^{(m)}$ that yielded this response is identified. The current object scale is then computed as the ratio $s_f = \frac{D_0}{D_t}$ between the depth of the target in the first frame (D_0), and the depth D_t at the current position. The depth is calculated by the median of the D channel within the target bounding box. The filter that yielded the best correlation response ($\mathbf{h}^{(m)}$) is correlated again on the search region scaled by s_f and target presence test is carried out (Section 3.3.1). In case the test determines the target is present, the current filter is replaced, i.e., $\mathbf{h}_t \leftarrow \mathbf{h}^{(m)}$, and the re-detection process is deactivated.

3.3.1 Target presence test

Recently, a target presence test has been proposed for long-term discriminative correlation filters [30]. The test is based on computing tracking uncertainty value as a ratio $q_t = \frac{R_t}{\overline{R}}$ between the maximum correlation response in the current frame (R_t) and a moving average of these values in the recent N_q frames when the target was visible. The test considers target lost whenever the ratio exceeds a pre-defined threshold $q_t > \tau_q$. It was showed in [30] that the test is robust to a range of thresholds.

To allow early occlusion detection, however, [19] introduce a test that compares the area of the segmentation mask with the area of the axis-aligned bounding box of the DCF. This test improves performance during occlusion, but gradual errors in scale estimation result in disagreement between the bounding box and the actual object and might lead to a reduced accuracy of the test.

The two tests are complementary and computationally very efficient, and the target presence is reported only if the considered target position passes the both tests.

3.4. Object tracking by reconstruction

Our object tracking by reconstruction approach (OTR) is summarized as follows.

Initialization. The tracker is initialized from a bounding box in the first frame. Color and depth histograms are sampled as in [19] and a segmentation mask $\mathbf{m}$ is generated. The segmentation mask $\mathbf{m}$ is used to learn the initial filter $\mathbf{h}_0$ according to (2), as well as to identify target pixels in the RGB-D model to initialize the pre-image Θ_0 by [33]. The set of snapshots is set to an empty set.

Localization. A tracking iteration at frame t starts with the target position $\mathbf{x}_{t-1}$ from the previous frame. A region is extracted around $\mathbf{x}_{t-1}$ in the current image and the position $\mathbf{x}_t$ with maximum correlation response is computed using the current filter $\mathbf{h}_{t-1}$ (along with all snapshots every N_{R} frames) as described in Section 3.2. The position $\mathbf{x}_t$ is tested using the target presence test from Section 3.3.1. If the test is passed, the target is considered as well localized, and the visual models (i.e., filters and pre-image) are updated. Otherwise, target re-detection (Section 3.3) is activated in the next frame.

Update. A color+depth segmentation mask $\mathbf{m}$ is computed within a region centered at $\mathbf{x}_t$ according to [19] to identify target pixels. The corresponding RGB-D pixels are used to update the pre-image Θ_t, i.e., the 3D surfel representation along with its 3D pose (Section 3.2.2).

The filter $\mathbf{h}_{t-1}$ is updated (3) by the filter learned at the current position (2) with support constraint computed from the pre-image (Section 3.2.1). Finally, the target aspect change is computed using the updated pre-image and the set of snapshots are updated if significant appearance change is detected (Section 3.2.3)

4. Experimental analysis

In this section, we validate OTR by a comprehensive experimental evaluation. The implementation details are pro-

Table 1. Experiments on the Princeton Tracking Benchmark using the PTB protocol. Numbers in the parenthesis are the ranks.

Method	*Avg.* Rank	*Avg.* Success	*Human*	*Animal*	*Rigid*	*Large*	*Small*	*Slow*	*Fast*	*Occ.*	*No-Occ.*	*Passive*	*Active*
OTR	**2.36**	**0.769(1)**	0.77(2)	0.68(6)	0.81(2)	0.76(4)	**0.77(1)**	0.81(2)	**0.75(1)**	0.71(3)	0.85(2)	**0.85(1)**	0.74(2)
ca3dms+toh [26]	4.55	0.737(5)	0.66(9)	0.74(2)	**0.82(1)**	0.73(7)	0.74(2)	0.80(4)	0.71(7)	0.63(9)	**0.88(1)**	0.83(2)	0.70(6)
CSR-rgbd++ [19]	5.00	0.740(3)	0.77(3)	0.65(8)	0.76(7)	0.75(5)	0.73(3)	0.80(3)	0.72(4)	0.70(4)	0.79(8)	0.79(6)	0.72(4)
3D-T [3]	5.64	0.750(2)	**0.81(1)**	0.64(9)	0.73(12)	**0.80(1)**	0.71(6)	0.75(9)	0.75(2)	**0.73(1)**	0.78(11)	0.79(7)	0.73(3)
PT [35]	6.09	0.733(6)	0.74(6)	0.63(11)	0.78(3)	0.78(3)	0.70(7)	0.76(5)	0.72(6)	0.72(2)	0.75(13)	0.82(4)	0.70(7)
OAPF [31]	6.09	0.731(7)	0.64(12)	**0.85(1)**	0.77(6)	0.73(8)	0.73(5)	**0.85(1)**	0.68(9)	0.64(8)	0.85(3)	0.78(9)	0.71(5)
DLST [1]	6.45	0.740(4)	0.77(4)	0.69(5)	0.73(13)	0.80(2)	0.70(9)	0.73(11)	0.74(3)	0.66(6)	0.85(4)	0.72(13)	**0.75(1)**
DM-DCF [20]	6.91	0.726(8)	0.76(5)	0.58(13)	0.77(5)	0.72(9)	0.73(4)	0.75(8)	0.72(5)	0.69(5)	0.78(10)	0.82(3)	0.69(9)
DS-KCF-Shape [16]	7.27	0.719(9)	0.71(7)	0.71(4)	0.74(9)	0.74(6)	0.70(8)	0.76(6)	0.70(8)	0.65(7)	0.81(6)	0.77(11)	0.70(8)
DS-KCF [6]	9.91	0.693(11)	0.67(8)	0.61(12)	0.76(8)	0.69(10)	0.70(10)	0.75(10)	0.67(11)	0.63(10)	0.78(12)	0.79(8)	0.66(10)
DS-KCF-CPP [16]	10.09	0.681(12)	0.65(10)	0.64(10)	0.74(10)	0.66(12)	0.69(12)	0.76(7)	0.65(12)	0.60(12)	0.79(9)	0.80(5)	0.64(12)
hiob-lc2 [36]	10.18	0.662(13)	0.53(13)	0.72(3)	0.78(4)	0.61(13)	0.70(11)	0.72(12)	0.64(13)	0.53(13)	0.85(5)	0.77(12)	0.62(13)
STC [40]	10.45	0.698(10)	0.65(11)	0.67(7)	0.74(11)	0.68(11)	0.69(13)	0.72(13)	0.68(10)	0.61(11)	0.80(7)	0.78(10)	0.66(11)

vided in Section 4.1. Performance analysis on two challenging RGB-D datasets, Princeton Tracking Benchmark (PTB) and STC, is reported in Section 4.2 and Section 4.3, respectively. Ablation studies are presented in Section 4.4 to verify our design choices.

4.1. Implementation details

We use HOG features [10] and colornames [37] in our tracker. The parameters related to the tracker are taken from [19]. The ICP error threshold is empirically set to $\tau_{\text{ICP}} = 5 \cdot 10^{-4}$ and the aspect ratio change threshold is set to $\tau_\rho = 0.20$. Maximum filter evaluation period is equal to $N_{\text{R}} = 5$ frames and $\alpha_s = 1.07$. All experiments are run on a single laptop with Intel Core i7 3.6GHz and the parameters for both tracking and 3D reconstruction are kept constant throughout the experiments. Our non-optimized implementation runs at 2 fps.

4.2. Performance on PTB benchmark [35]

The Princeton Tracking Benchmark [35] is the most comprehensive and challenging RGB-D tracking benchmark to date. The authors have recorded and manually annotated 100 RGB-D videos in real-life conditions using a Kinect v1.0. Ground truth bounding boxes of five sequences are publicly available whereas the ground truth for the remaining 95 sequences are kept hidden to prevent overfitting. Tracking performance is evaluated on the 95 sequences with the hidden ground-truth. The sequences are grouped into 11 categories: *Human*, *Animal*, *Rigid*, *Large*, *Small*, *Slow*, *Fast*, *Occlusion*, *No Occlusion*, *Passive* and *Active*. We use Bibi *et al*. [3] protocol with improved depth registration in the experiments.

The performance is measured by employing a PASCAL VOC [13] type of evaluation. Per-frame overlap o_t is defined as

$$o_t = \begin{cases} \frac{area(B_{TR} \cap B_{GT})}{area(BTR \cup B_{GT})}, & \text{if both } B_{TR} \text{ and } B_{GT} \text{ exist} \\ 1, & \text{if neither } B_{TR} \text{ and } B_{GT} \text{ exists} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where B_{TR} is the output bounding box of the tracker and B_{GT} is the ground truth bounding box. Tracking performance is given as *success rate* which represents average overlap [7]. The PTB evaluation protocol sorts the trackers according to the primary performance measures with respect to each object category and computes the final ranking as the average over these ranks. In addition, the overall success rate is reported for detailed analysis.

The OTR tracker is compared to all trackers available on the PTB leaderboard: ca3dms+toh [26], CSR-rgbd++ [19], 3D-T [3], PT [35], OAPF [31], DM-DCF [20], DS-KCF-Shape [16], DS-KCF [6], DS-KCF-CPP [16], hiob_lc2 [36] and we added two recent trackers STC [40] and DLST [1]. Results are reported in Table 1.

OTR convincingly sets the new *state-of-the-art* in terms of both overall ranking and the average success by a large margin compared to the next-best trackers (Table 1). In terms of average success, OTR obtains a 4.3% gain compared to the second ranking tracker ca3dms+toh [26], which tracks the target in 3D as well, but without reconstruction. This result speaks in favour of our 3D-based pre-image construction and its superiority for RGB-D tracking.

In addition to being the top overall tracker, the performance of OTR is consistent across all categories. OTR is consistently among the top trackers in each category and achieves the top rank in three categories and the second best in five categories. This suggests that our tracker does not overfit to a certain type of scenario and it generalizes very well unlike some other methods in the benchmark.

A closely related work to our own is recent CSR-rgbd++, which combines a single CSR-DCF with color and depth segmentation and implements a target re-detection. OTR obtains a significant 6.6% increase over CSR-rgbd++ in *Rigid* category, which speaks in favor of our DCFs approach with several views connected to a 3D pre-image that localizes the target more precisely. On the *No-Occ.* category, OTR outperforms CSR-rgbd++ by a 7.6% success rate. This can be attributed to the advantage of using a pre-image Θ for DCF training described in Section 3.2.1.

Table 2. The normalized area under the curve (AUC) scores computed from one-pass evaluation on the STC Benchmark [40].

	Attributes										
Method	*AUC*	*IV*	*DV*	*SV*	*CDV*	*DDV*	*SDC*	*SCC*	*BCC*	*BSC*	*PO*
OTR	**0.49**	**0.39**	**0.48**	**0.31**	0.19	**0.45**	**0.44**	0.46	**0.42**	**0.42**	**0.50**
CSR-rgbd++ [19]	0.45	0.35	0.43	0.30	0.14	0.39	0.40	0.43	0.38	0.40	0.46
ca3dms+toh [26]	0.43	0.25	0.39	0.29	0.17	0.33	0.41	**0.48**	0.35	0.39	0.44
STC [40]	0.40	0.28	0.36	0.24	**0.24**	0.36	0.38	0.45	0.32	0.34	0.37
DS-KCF-Shape [16]	0.39	0.29	0.38	0.21	0.04	0.25	0.38	0.47	0.27	0.31	0.37
PT [35]	0.35	0.20	0.32	0.13	0.02	0.17	0.32	0.39	0.27	0.27	0.30
DS-KCF [6]	0.34	0.26	0.34	0.16	0.07	0.20	0.38	0.39	0.23	0.25	0.29
OAPF [31]	0.26	0.15	0.21	0.15	0.15	0.18	0.24	0.29	0.18	0.23	0.28

4.3. Performance on STC benchmark [40]

The STC benchmark [40] has been recently published to complement the PTB benchmark in the number of categories and diversity of sequences. 36 sequences are recorded indoors and outdoors using Asus Xtion sensors and the authors annotated every frame of every video with 10 attributes; *Illumination variation* (IV), *Depth variation* (DV), *Scale variation* (SV), *Color distribution variation* (CDV), *Depth distribution variation* (DDV), *Surrounding depth clutter* (SDC), *Surrounding color clutter* (SCC), *Background color camouflages* (BCC), *Background shape camouflages* (BSC), *Partial occlusion* (PO). These attributes were either automatically computed or manually annotated.

The tracking performance is measured by precision and success plots computed from a one-pass evaluation akin to [39]. Success plot shows the portion of correctly tracked frames with respect to the different values of the overlap thresholds. Tracking performance is measured by a non-normalized area under the curve on this graph, i.e., the sum of values on the plot. The standard AUC measure [39] is obtained by dividing the non-normalized AUC by the number of overlap thresholds. The number of thresholds is the same for all evaluated trackers and only scales the non-normalized AUC to interval $[0, 1]$. We therefore report the standard AUC values, which is the more familiar measure in the tracking community. Precision plot is constructed similarly to success plot, by measuring the portion of frames with center-error smaller than a threshold. The overall measure on precision plot is computed as the value at 20 pixels error threshold.

The OTR tracker is compared to the following trackers: CSR-rgbd++ [19], ca3dms+toh [26], STC [40], DS-KCF-Shape [16], PT [35], DS-KCF [6] and OAPF [31]. The results are presented in Table 2 and Figure 4. As on PTB benchmark (Section 4.2), OTR outperforms the *state-of-the-art* by a large margin not only in the overall score but in most of the categories except *CDV (Color Distribution Variation)* and *SCC (Surrounding Color Clutter)*, where it is ranked among top three trackers. The overall top performance and excellent per-attribute performance support our observations on PTB benchmark that OTR is capable of handling various tracking scenarios and generalizes well over the different datasets. Qualitative tracking results on the four sequences from STC dataset are shown in Figure 5. The computing times for the three best performing trackers are 2 fps (OTR), 6 fps (CSR-rgbd++) and 34 fps (ca3dms+toh).

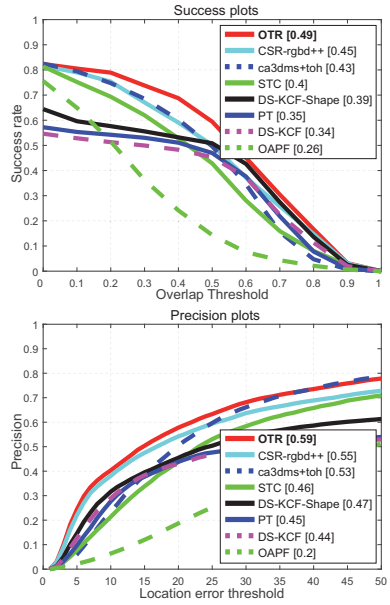

Figure 4. Success and precision plots on STC benchmark [40].

4.4. Ablation studies

The main components of our tracker are (i) the 3D-based pre-image, which provides an improved target segmentation, (ii) the set of multiple view-specific target DCFs and (iii) the interaction between the former two components. An ablation study is conducted on the PTB [35] dataset to evaluate the extent of contribution of each component. We implemented three variants of the proposed tracker with the

Table 3. Ablation studies on the PTB benchmark [35].

Method	*Avg.* Success	*Human*	*Animal*	*Rigid*	*Large*	*Small*	*Slow*	*Fast*	*Occ.*	*No-Occ.*	*Passive*	*Active*
OTR	**0.769**	0.77	**0.68**	**0.81**	**0.76**	**0.77**	**0.81**	**0.75**	**0.71**	**0.85**	**0.85**	**0.74**
OTR_{-3D}	0.747	0.76	0.62	0.80	0.75	0.75	0.80	0.72	0.69	0.82	0.84	0.71
OTR_{-VS}	0.743	0.75	0.66	0.77	0.74	0.74	0.79	0.72	0.67	0.84	0.81	0.72
OTR_{-3D-VS}	0.740	**0.78**	0.61	0.76	0.75	0.73	0.79	0.72	**0.71**	0.78	0.79	0.72

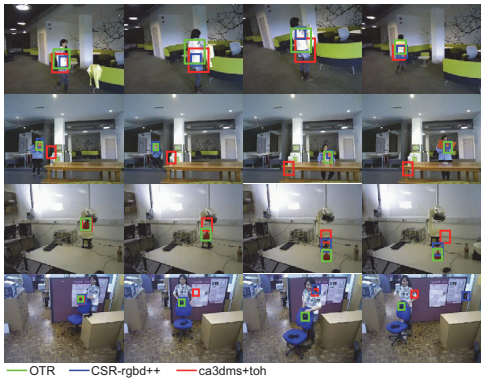

Figure 5. Tracking results on four sequences from STC dataset [40]. The proposed OTR tracker confidently tracks the target undergoing a substantial pose change. Two state-of-the-art RGB-D trackers (CSR-rgbd++ [19] and cs3dms+toh [26]), that do not apply the multi-view DCFs nor target 3D pre-image, result in less accurate localization or failure.

3D pre-image and view-specific DCFs. The first variant is the tracker without the 3D pre-image, denoted as OTR_{-3D}. The second variant is the tracker without the view-specific DCFs (OTR_{-VS}) and the third variant is the tracker without the view-specific DCFs and without the 3D pre-image (OTR_{-3D-VS}).

The results of the ablation study are reported in Table 3. The proposed OTR with all components achieves a 0.769 success rate. Removing the view-specific target representation (OTR_{-VS}) or 3D pre-image (OTR_{-3D}) result to approx. 3% success rate drop in tracking performance (0.747 and 0.743). Removing both view-specific and 3D pre-image representation (OTR_{-3D-VS}) further reduces the tracking performance to 0.740 success rate.

On the *Occlusion* category the OTR tracker outperforms the version without a view-specific formulation (OTR_{-VS}) by 6% increase in the success rate. The view-specific set of DCFs *remembers* the target appearance from different views, which helps in reducing drifting and improves re-detection accuracy after occlusion. On average, 4 views were automatically generated by the view-specific DCF in OTR per tracking sequence. The tracker version without the view-specific formulation *forgets* the past appearance, which reduces the re-detection capability.

In situations without occlusion, the 3D pre-image plays a more important role than the view-specific DCF formulation. Removing the 3D pre-image creation from the tracker results in 7% success rate reduction, which indicates the significant importance of using the 3D pre-image for robust DCF learning.

Overall, the addition of 3D pre-image and view-specific target representation improves performance of the baseline version OTR_{-3D-VS} by approximately 4% in tracking success rate. The ablation study results conclusively show that every component importantly contributes to the tracking performance boost.

5. Conclusions

A new long-term RGB-D tracker, called OTR – Object Tracking by Reconstruction is presented. The target 3D model, a pre-image, is constructed by a surfel-based ICP. The limited convergence range of the ICP and the requirement to automatically identify object pixels used for reconstruction is addressed by utilizing a DCF for displacement estimation and for approximate target segmentation. The 3D pre-image in turn constrains the DCF learning, and is used for generating view-specific DCFs. These are used for localization as well as for target re-detection, giving the tracker a long-term tracking quality.

The OTR tracker is extensively evaluated on two challenging RGB-D tracking benchmarks and compared to 12 *state-of-the-art* RGB-D trackers. OTR outperforms all trackers by a large margin, setting a new *state-of-the-art* on these benchmarks. An ablation study verifies that the performance improvements come from the 3D pre-image construction, the view-specific DCF set and the interaction between the two.

The view-specific DCF formulation allows long-term tracking of poorly textured and small objects over large displacements. Our future work will focus on extension to model-based tracking with pre-learned models on realistic, open-world scenarios. In addition, we plan to consider improvements by ICP robustification and deep features.

Acknowledgments

This work is supported by Business Finland under Grant 1848/31/2015 and Slovenian research agency program P2-0214 and project J2-8175. J. Matas is supported by the Technology Agency of the Czech Republic project TE01020415 – V3C Visual Computing Competence Center.

References

[1] N. An, X.-G. Zhao, and .-G. Hou. Online RGB-D Tracking via Detection-Learning-Segmentation. In *ICPR*, 2016.

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-Convolutional Siamese Networks for Object Tracking. In *ECCV Workshops*, 2016.

[3] A. Bibi, T. Zhang, and B. Ghanem. 3D Part-Based Sparse Tracker with Automatic Synchronization and Registration. In *CVPR*, 2016.

[4] D. S. Bolme, J.R. Beveridge, B. A. Draper, and Y.-M. Lui. Visual Object Tracking using Adaptive Correlation Filters. In *CVPR*, 2010.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[6] M. Camplani, S. Hannuna, M. Mirmehdi, D. Damen, A. Paiement, L. Tao, and T. Burghardt. Real-time RGB-D Tracking with Depth Scaling Kernelised Correlation Filters and Occlusion Handling. In *BMVC*, 2015.

[7] L. Čehovin, A. Leonardis, and M. Kristan. Visual Object Tracking Performance Measures Revisited. *IEEE TIP*, 25(3):1261–1274, 2016.

[8] F. Chaumette, P. Rives, and B. Espiau. Positioning of a Robot with Respect to an Object, Tracking it and Estimating its Velocity by Visual Servoing. In *ICRA*, 1991.

[9] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. *IEEE PAMI*, 25:564–567, 2003.

[10] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[11] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Discriminative Scale Space Tracking. *IEEE PAMI*, 39(8):1561–1575, 2017.

[12] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer. Adaptive Color Attributes for Real-Time Visual Tracking. In *CVPR*, 2014.

[13] M. Everingham, L. van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010.

[14] H.K. Galoogahi, T. Sim, and S. Lucey. Correlation Filters with Limited Boundaries. In *CVPR*, 2015.

[15] H. K. Galoogahi, T. Sim, and S. Lucey. Multi-channel Correlation Filters. In *ICCV*, 2013.

[16] S. Hannuna, M. Camplani, J. Hall, M. Mirmehdi, D. Damen, T. Burghardt, A. Paiement, and L. Tao. DS-KCF: A Real-time Tracker for RGB-D Data. *Journal of Real-Time Image Processing*, 2016.

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-Speed Tracking with Kernelized Correlation Filters. *IEEE PAMI*, 37(3):583–596, 2015.

[18] W. Hu, T. Tan, L. Wang, and S. Maybank. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3):334–352, 2004.

[19] U. Kart, J.-K. Kämäräinen, and J. Matas. How to Make an RGBD Tracker ? In *ECCV Workshops*, 2018.

[20] U. Kart, J.-K. Kämäräinen, J. Matas, L. Fan, and F. Cricri. Depth Masked Discriminative Correlation Filter. In *ICPR*, 2018.

[21] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, 2007.

[22] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojír, and et al. The Visual Object Tracking VOT2016 Challenge Results. In *ECCV Workshops*, 2016.

[23] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, and et al. The Visual Object Tracking VOT2017 Challenge Results. In *ICCV Workshops 2017*.

[24] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojír, and et al. The Visual Object Tracking VOT2014 Challenge Results. In *ECCV Workshops*, 2014.

[25] K. Lebeda, S. Hadfield, and R. Bowden. 2D Or Not 2D: Bridging the Gap Between Tracking and Structure from Motion. In *ACCV*, 2014.

[26] Y. Liu, X.-Y. Jing, J. Nie, H. Gao, J. Liu, and G.-P. Jiang. Context-aware 3-D Mean-shift with Occlusion Handling for Robust Object Tracking in RGB-D Videos. *IEEE TMM*, 2018.

[27] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, 1981.

[28] A. Lukežič, T. Vojír, L. Čehovin, J. Matas, and M. Kristan. Discriminative Correlation Filter with Channel and Spatial Reliability. In *CVPR*, 2017.

[29] A. Lukežič, T. Vojír, L. Čehovin Zajc, J. Matas, and M. Kristan. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. *IJCV*, 2018.

[30] A. Lukežič, L. Čehovin Zajc, T. Vojír, J. Matas, and M. Kristan. FCLT - A Fully-Correlational Long-Term Tracker. In *ACCV*, 2018.

[31] K. Meshgi, S. Maeda, S. Oba, H. Skibbe, Y. Li, and S. Ishii. An Occlusion-aware Particle Filter Tracker to Handle Complex and Persistent Occlusions. *CVIU*, 150:81 – 94, 2016.

[32] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011.

[33] M. Rünz and L. Agapito. Co-Fusion: Real-time Segmentation, Tracking and Fusion of Multiple Objects. In *ICRA*, 2017.

[34] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual Tracking: An Experimental Survey. *IEEE PAMI*, 36(7):1442–1468, 2014.

[35] S. Song and J. Xiao. Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. In *ICCV*, 2013.

[36] P. Springstübe, S. Heinrich, and S. Wermter. Continuous Convolutional Object Tracking. In *ESANN*, 2018.

[37] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning Color Names for Real-world Applications. *IEEE TIP*, 18(7):1512–1523, 2009.

[38] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. Elasticfusion. *Int. J. Rob. Res.*, 35(14):1697–1716, 2016.

[39] Y. Wu, J. Lim, and Y. Ming-Hsuan. Object Tracking Benchmark. *IEEE PAMI*, 37:1834 – 1848, 2015.

[40] J. Xiao, R. Stolkin, Y. Gao, and A. Leonardis. Robust Fusion of Color and Depth Data for RGB-D Target Tracking Using Adaptive Range-Invariant Depth Models and Spatio-Temporal Consistency Constraints. *IEEE Transactions on Cybernetics*, 48:2485 – 2499, 2018.

PUBLICATION

IV

CDTB: A Color and Depth Visual Object Tracking Dataset and Benchmark

A. Lukežič, U. Kart, J. Käpylä, A. Durmush, J.-K. Kämäräinen, J. Matas and M. Kristan

*IEEE/CVF International Conference on Computer Vision (ICCV)*2019

CDTB: A Color and Depth Visual Object Tracking Dataset and Benchmark

Alan Lukežič[1], Ugur Kart[2], Jani Käpylä[2], Ahmed Durmush[2], Joni-Kristian Kämäräinen[2], Jiří Matas[3] and Matej Kristan[1]

[1]Faculty of Computer and Information Science, University of Ljubljana, Slovenia

[2]Computing Sciences, Tampere University, Finland

[3]Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

alan.lukezic@fri.uni-lj.si

Abstract

We propose a new color-and-depth general visual object tracking benchmark (CDTB). CDTB is recorded by several passive and active RGB-D setups and contains indoor as well as outdoor sequences acquired in direct sunlight. The CDTB dataset is the largest and most diverse dataset for RGB-D tracking, with an order of magnitude larger number of frames than related datasets. The sequences have been carefully recorded to contain significant object pose change, clutter, occlusion, and periods of long-term target absence to enable tracker evaluation under realistic conditions. Sequences are per-frame annotated with 13 visual attributes for detailed analysis. Experiments with RGB and RGB-D trackers show that CDTB is more challenging than previous datasets. State-of-the-art RGB trackers outperform the recent RGB-D trackers, indicating a large gap between the two fields, which has not been detected by the prior benchmarks. Based on the results of the analysis we point out opportunities for future research in RGB-D tracker design.

1. Introduction

Visual object tracking has been enjoying a significant interest of the research community for over several decades due to scientific challenges it presents and its large practical potential. In its most general formulation, it addresses localization of an arbitrary object in all frames of a video, given a single annotation specified in one frame. This is a challenging task of self-supervised learning, since a tracker has to localize and carefully adapt to significant target appearance changes, cope with ambient changes, clutter, and detect occlusion and target disappearance. As such, general object trackers cater a range of applications and research challenges like surveillance systems, video editing, sports analytics and autonomous robotics.

Fuelled by emergence of tracking benchmarks [40, 44, 27, 25, 37, 36] that facilitate objective comparison of different approaches, the field has substantially advanced in the last decade. Due to a wide adoption of RGB cameras, the benchmarks have primarily focused on color (RGB) trackers and trackers that combine color and thermal (infrared) modalities [28, 26, 23, 24].

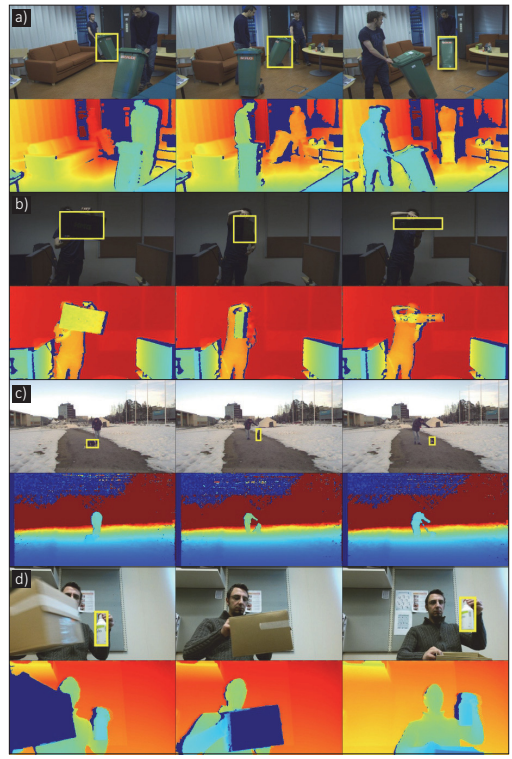

Figure 1. RGB and depth sequences from CDTB. Depth offers a complementary information to color: two identical objects are easier to distinguish in depth (a), low illumination scenes (b) are less challenging for trackers if depth information is available, tracking a deformable object in depth simplifies the problem (c) and a sudden significant change in depth is a strong clue for occlusion (d). Sequences (a,b) are captured by a ToF-RGB pair of cameras, (c) by s tereo-camera sensor and (d) by a Kinect sensor.

Only recently various depth sensors like RGB-D, time-

of-flight (ToF) and LiDAR have become widely accessible. Depth provides an important cue for tracking since it simplifies reasoning about occlusion and offers a better object-to-background separation compared to only color. In addition, depth is a strong cue to acquire object 3D structure and 3D pose without a prior 3D model, which is crucial in research areas like robotic manipulation [5]. The progress in RGB-D tracking has been boosted by the emergence of RGB-D benchmarks [41, 45], but the field significantly lags behind the advancements made in RGB-only tracking.

One reason for the RGB – RGB-D general object tracking performance gap is that existing RGB-D benchmarks [41, 45] are less challenging than their RGB counterparts. The sequences are relatively short from the perspective of practical applications, the objects never leave and re-enter the field of view, they undergo only short-term occlusions and rarely significantly rotate away from the camera. The datasets are recorded indoor only with Kinect-like sensors which prohibits generalization of the results to general outdoor setups. These constraints were crucial for early development of the field, but further boosts require a more challenging benchmark, which is the topic of this paper.

In this work we propose a new color-and-depth tracking benchmark (CDTB) that makes several contributions to the field of general object RGB-D tracking. (i) The CDBT dataset is recorded by several color-and-depth sensors to capture a wide range of depth signals. (ii) The sequences are recorded indoor as well as outdoor to extend the domain of tracking setups. (iii) The dataset contains significant object pose changes to encompass depth appearance variability from real-world tracking environment. (iv) The objects are occluded or leave the field of view for longer duration to emphasize the importance of trackers being able to report target loss and perform re-detection. (v) We compare several state-of-the-art RGB-D trackers as well as state-of-the-art RGB trackers and their RGB-D extensions. Examples of CDTB dataset are shown in Figure 1.

The reminder of the paper is structured as follows. Section 2 summarizes the related work, Section 3 details the acquisition and properties of the dataset, Section 4 summarizes the performance measures, Section 5 reports experimental results and Section 6 concludes the paper.

2. Related work

RGB-D Benchmarks. The diversity of the RGB-D datasets is limited compared to those in RGB tracking. Many of the datasets are application specific, e.g., *pedestrian tracking* or *hand tracking*. For example, Ess *et al*. [11] provide five 3D bounding box annotated sequences captured by a calibrated stereo-pair, the RGB-D People Dataset [42] contains a single sequence of pedestrians in a hallway captured by a static RGB-D camera and Stanford Office [8] contains 17 sequences with a static and one with a moving Kinect. Garcia-Hernando *et al*. [13] introduce an RGB-D dataset for hand tracking and action recognition. Another important application field for RGB-D cameras is robotics, but here datasets are often small and the main objective is real-time model-based 3D pose estimation. For example, the RGB-D Object Pose Tracking Dataset [7] contains 4 synthetic and 2 real RGB-D image sequences to benchmark visual tracking and 6-DoF pose estimation. Generating synthetic data has become popular due to requirements of large training sets for deep methods [39], but it is unclear how well these predict real world performance.

Only two datasets are dedicated to general object tracking. The most popular is Princeton Tracking Benchmark (PTB) [41], which contains 100 RGB-D video sequences of rigid and nonrigid objects recorded with Kinect. The choice of sensor constrains the dataset to only indoor scenarios. The dataset diversity is further reduced since many sequences share the same tracked objects and the background. More than half of the sequences are people tracking. The sequences are annotated by five global attributes. The RGB and depth channels are poorly calibrated. In approximately 14% of sequences the RGB and D channels are not synchronized and approximately 8% are miss-aligned. The calibration issues were addressed by Bibi et al [3] who published a corrected dataset. PTB addresses long-term tracking, in which the tracker has to detect target loss and perform re-detection. The dataset thus contains several full occlusions, but the target never leaves and re-enters the field of view, thus limiting the evaluation capabilities of re-detecting trackers. Performance is evaluated as the percentage of frames in which the bounding box predicted by tracker exceeds a 0.5 overlap with the ground truth. The overlap is artificially set to 1 when the tracker accurately predicts target absence. Recent work in long-term tracker performance evaluation [43, 33] argue against using a single threshold and [33] further show reduced interpretation strength of the measure used in PTB.

The Spatio-Temporal Consistency dataset (STC) [45] was recently proposed to address the drawbacks of PTB. The dataset is recorded by Asus Xtion RGB-D sensor, which also constrains the dataset to only indoor scenarios and a few low-light outside scenarios, but care has been taken to increase the sequence diversity. The dataset is smaller than PTB, containing only 36 sequences, but annotated by thirteen global attributes. STC addresses short-term tracking scenario, i.e., trackers are not required to perform re-detection. Thus the sequences are relatively short and the short-term performance evaluation methodology is used. This makes the dataset inappropriate for evaluating trackers useful in many practical setups, in which target loss detection and redetection are crucial capabilities.

RGB Trackers. Recent years have seen a surge in Short-term Trackers (ST) and especially Discriminative Correla-

tion Filter (DCF) based approaches have been popular due to their mathematical simplicity and elegance. In their seminal paper, Bolme *et al*. [4] proposed using DCF for visual object tracking. Henriques *et al*. [16] proposed an efficient training method by exploiting the properties of circular convolution. Lukezic *et al*. [32] and Galoogahi *et al*. [12] proposed a mechanism to handle boundary problems and segmentation-based DCF constraints have been introduced in [32]. Danelljan *et al*. [10] used a factorized convolution operator and achieved excellent scores on well-known benchmarks.

As a natural extension of the ST, Long-term Trackers (LT) have been proposed [18] where the tracking is decomposed into short-term tracking and long-term detection. Lukezic *et al*. proposed a fully-correlational LT [31] by storing multiple correlation filters that are trained at different time scales. Zhang *et al*. [46] used deep regression and verification networks and they achieved the top rank in VOT-LT 2018 [25]. Despite being published as an ST, MDNet [38] has proven itself as an efficient LT. MDNet uses discriminatively trained Convolutional Neural Networks(CNN) and won the VOT 2015 challenge [26].

RGB-D Trackers. Compared to RGB trackers, the body of literature on RGB-D trackers is rather limited which can be attributed to the lack of available datasets until recently. In 2013, the publication of PTB [41] ignited the interest in the field and there have been numerous attempts by adopting different approaches. The authors of PTB have proposed multiple baseline trackers which use different combinations of HOG [9], optical flow and point clouds. As a part of particle filter tracker family, Meshgi *et al*. [34] proposed a particle filter framework with occlusion awareness using a latent occlusion flag. They pre-emptively predict the occlusions, expand the search area in case of occlusions. Bibi *et al*. [3] represented the target by sparse, part-based 3-D cuboids while adopting particle filter as their motion model. Hannuna *et al*. [14], An *et al*. [1] and Camplani *et al*. [6] extended the Kernelized Correlation Filter (KCF) RGB tracker [16] by adding the depth channel. Hannuna *et al*. and Camplani *et al*. proposed a fast depth image segmentation which is later used for scale, shape analysis and occlusion handling. An *et al*. proposed a framework where the tracking problem is divided into detection, learning and segmentation. To use depth inherently in DCF formulation, Kart *et al*. [20] adopted Gaussian foreground masks on depth images in CSRDCF [32] training. They later extended their work by using a graph cut method with color and depth priors for the foreground mask segmentation [19] and more recently proposed a view-specific DCF using object's 3D structure based masks [21]. Liu *et al*. [30] proposed a 3D mean-shift tracker with occlusion handling. Xiao *et al*. [45] introduced a two-layered representation of the target by adopting a spatio-temporal consistency constraints.

3. Color and depth tracking dataset

We used several RGB-D acquisition setups to increase the dataset diversity in terms of acquisition hardware. This allowed unconstrained indoor as well as outdoor sequence acquisition, thus diversifying the dataset and broaden the scope of scenarios from real-word tracking environment. The following three acquisition setups were used: (i) RGB-D sensor (Kinect), (ii) time-of-flight (ToF)-RGB pair and (iii) stereo cameras pair. The setups are described in the following.

RGB-D Sensor sequences were captured with a Kinect v2 that outputs 24-bit 1920×1080 RGB images (8-bit per color channel) and 512×424 32-bit floating point depth images with an average frame rate of 30 fps. JPEG compression is applied to RGB frames while depth data is converted into 16-bit unsigned integer and saved in PNG format. The RGB and depth images are synchronized internally and no further synchronization was required.

ToF-RGB pair consists of Basler tof640-20gm time-of-flight and Basler acA1920-50gc color cameras. The ToF camera has 640x480pix resolution and maximum 20 fps frame rate whereas color camera has 1920x1200pix resolution and 50 fps maximum frame rate at full resolution. Both cameras can be triggered externally using the I/O's of the cameras for external synchronisation. The cameras were mounted on a high precision CNC-machined aluminium base in a way that the baseline of the cameras are 75.2mm and camera sensor center points are on the same level. The TOF camera has built in optics with $57^\circ \times 43^\circ$ (HxV) field-of-view. The color camera was equipped with a 12mm focal length lens (VS-1214H1), which has $56.9^\circ \times 44^\circ$ (HxV) field-of-view for 1" sensors, to match the field-of-view of the ToF camera. The cameras were synchronised by an external triggering device at the rate of 20 fps. The color camera output was 8-bit raw Bayer images whereas ToF camera output was 16-bit depth images. The raw Bayer images were later debayered to 24-bit RGB images (8-bit per color channel).

Stereo-cameras pair is composed of two Basler acA1920-50gc color cameras which are mounted on a high precision machined aluminium base with 70mm baseline. The cameras were equipped with 6mm focal length lenses (VS-0618H1) with $98.5^\circ \times 77.9^\circ$ (HxV) field-of-view for 1" sensors. The cameras were synchronised by an external triggering device at the rate of 40 fps at full resolution. The camera outputs were 8-bit raw Bayer images which were later Bayer demosaiced to 24-bit RGB images (8-bit per color channel). A semi-global block matching algorithm [17] was applied to the rectified stereo images and converted to metric depth values using the camera calibration parameters.

3.1. RGB and Depth Image Alignment

All three acquisition setups were calibrated using the Caltech Camera Calibration Toolbox[1] with standard modifications to cope with image pairs of different resolution for the RGB-D sensor and ToF-RGB-pair setups. The calibration provides the external camera parameters, *rotation matrix* $\boldsymbol{R}_{3\times3}$ and *translation vector* $\boldsymbol{t}_{3\times1}$, and the intrinsic camera parameters, *focal length* $\boldsymbol{f}_{2\times1}$, *principal point* $\boldsymbol{c}_{2\times1}$, *skew* α and lens distortion coefficients $\boldsymbol{k}_{5\times1}$. The forward projection is defined by [15]

$$\boldsymbol{m} = \mathcal{P}(\boldsymbol{x}) = (\mathcal{P}_c \circ \mathcal{R})(d), \tag{1}$$

where $\boldsymbol{x} = (x, y, z)^T$ is the scene point in world coordinates, $\boldsymbol{m}$ is the projected point in image coordinates and $d = I_{depth}(\boldsymbol{m})$ is the depth. $\mathcal{R}$ is a rigid Euclidean transformation, $\boldsymbol{x}_c = \mathcal{R}(\boldsymbol{x})$, defined by $\boldsymbol{R}$ and $\boldsymbol{t}$, and $\mathcal{P}_c$ is the intrinsic operation $\mathcal{P}_c(\boldsymbol{x}_c) = (\mathcal{K} \circ \mathcal{D} \circ \hat{\nu})(\boldsymbol{x}_c)$ of the perspective division operation $\hat{\nu}$, distortion operation $\mathcal{D}$ using $\boldsymbol{k}$ and the affine mapping $\mathcal{K}$ of $\boldsymbol{f}$ and α.

The depth images of RGB-D Sensor and ToF-RGB pair were per-pixel aligned to the RGB images as follows. A 3D point corresponding to each pixel in the calibrated depth image was computed using the inverse of (1) as $\boldsymbol{x} = \mathcal{P}^{-1}(\boldsymbol{m}, d)$. These points were projected to the RGB image and a linear interpolation model was used to estimate missing per-pixel-aligned re-projected depth values. For further studies we provide the original data and calibration parameters upon request.

3.2. Sequence Annotation

The VOT Aibu image sequence annotator[2] was used to manually annotate the targets by axis-aligned bounding boxes. The bounding boxes were placed following the VOT [28] definition by maximizing the number of target pixels within the bounding box and minimizing their number outside the bounding box. All bounding boxes were checked by several annotators for quality control. In case of a disagreement the authors consolidated and reached an agreement on annotation.

All sequences were annotated per-frame with thirteen attributes. We selected standard attributes for short-term tracking (partial occlusion, deformable target, similar targets, out-of-plane rotation, fast motion and target size change) and for the long-term tracking (target out-of-view and full occlusion). We additionally included RGBD tracking-specific attributes (reflective target, dark scene and depth change). The following attributes were manually annotated: (i) target out-of-view, (ii) full occlusion, (iii) partial occlusion, (iv) out-of-plane rotation, (v) similar objects, (vi) deformable target, (vii) reflective target and (viii) dark scene. The attribute (ix) fast motion was assigned to a frame in which the target center moves by at least 30% of its size in consecutive frames, (x) target size change was assigned when the ratio between maximum and minimum target size in 21 consecutive frames[3] was larger than 1.5 and (xi) aspect ratio change was assigned when the ratio between the maximum and minimum aspect (i.e., width / height) within 21 consecutive frames was larger than 1.5. The attribute (xii) depth change was assigned when the ratio between maximum and minimum of median of depth within target region in 21 consecutive frames was larger than 1.5. Frames not annotated with any of the first twelve attributes were annotated as (xiii) unassigned.

[1] http://www.vision.caltech.edu/bouguetj/calib_doc

[2] https://github.com/votchallenge/aibu

4. Performance Evaluation Measures

Tracker evaluation in a long-term tracking scenario in which targets may disappear/re-appear, requires measuring the localization accuracy, as well as re-detection capability and ability to report that target is not visible. To this end we adopt the recently proposed long-term tracking evaluation protocol from [33], which is used in the VOT2018 long-term challenge [25]. The tracker is initialized in the first frame and left to run until the end of the sequence without intervention.

The implemented performance measures are tracking precision (Pr) and recall (Re) from [33]. Tracking precision measures the accuracy of target localization when deemed visible, while tracking recall measures the accuracy of classifying frames with target visible. The two measures are combined into F-measure, which is the primary measure. In the following we briefly present how the measures are calculated. For details and derivation we refer the reader to [33].

We denote G_t as a ground-truth target pose and $A_t(\tau_\theta)$ as a pose prediction given by a tracker at frame t. The evaluation protocol requires that the tracker reports a confidence value besides the pose prediction. The confidence of the tracker in frame t is denoted as θ_t while confidence threshold is denoted as τ_θ. If the target is not visible in frame t, then ground-truth is an empty set i.e., $G_t = \emptyset$. Similarly, if tracker does not report the prediction or if confidence score is below the confidence threshold, i.e., $\theta_t < \tau_\theta$, then the output is an empty set $A_t(\tau_\theta) = \emptyset$.

From the object detection literature, when intersection-over-union between the tracker prediction and ground-truth $\Omega(A_t(\tau_\theta), G_t)$, exceeds overlap threshold τ_Ω, the prediction is considered as correct. This definition of correct prediction highly depends on the minimal overlap threshold τ_Ω. The problem is in [33] addressed by integrating tracking precision and recall over all possible overlap thresholds

[3] We observed that target size and aspect ratio change are reliably detected differentiating values at 10 frames before and after the current timestep - thus the discrete temporal derivative considers 21 frames.

which results in the following measures

$$Pr(\tau_\theta) = \frac{1}{N_p} \sum_{t \in \{t: A_t(\tau_\theta) \neq \emptyset\}} \Omega(A_t(\tau_\theta), G_t), \quad (2)$$

$$Re(\tau_\theta) = \frac{1}{N_g} \sum_{t \in \{t: G_t \neq \emptyset\}} \Omega(A_t(\tau_\theta), G_t), \quad (3)$$

where N_g is number of frames where target is visible, i.e., $G_t \neq 0$ and N_p is number of frames where tracker made a prediction, i.e., $A_t(\tau_\theta) \neq \emptyset$. Tracking precision and recall are combined into a single score by computing tracking F-measure $F(\tau_\theta) = \Big(2Re(\tau_\theta)Pr(\tau_\theta)\Big) / \Big(Re(\tau_\theta) + Pr(\tau_\theta)\Big)$. Tracking performance is visualized on precision-recall and F-measure plots by computing scores for all confidence thresholds τ_θ. The highest F-measure on the F-measure plot represents the optimal confidence threshold and it is used for ranking trackers. This process also does not require manual threshold setting for each tracker separately.

The performance measures are directly extended to per-attribute analysis. In particular, the tracking Precision, Recall and F-measure are computed from predictions on the frames corresponding to a particular attribute.

5. Experiments

This section presents experimental results on the CDTB dataset. Section 5.1 summarizes the list of tested trackers, Section 5.2 compares the CDTB dataset with most related datasets, Section 5.3 reports overall tracking performance and Section 5.4 reports per-attribute performance.

5.1. Tested Trackers

The following 16 trackers were chosen for evaluation. We tested (i) RGB baseline and state-of-the-art short-term correlation and deep trackers (KCF [16], NCC [29], BACF [22], CSRDCF [32], SiamFC [2], ECOhc [10], ECO [10] and MDNet [38]), (ii) RGB state-of-the-art long-term trackers (TLD [18], FuCoLoT [31] and MBMD [46]) and (iii) RGB-D state-of-the-art trackers (OTR [21] and Ca3dMS [30]). Additionally, the following RGB trackers have been modified to use depth information: ECOhc-D [19], CSRDCF-D [19] and KCF-D[4].

5.2. Comparison with Existing Benchmarks

Table 1 compares the properties of CDTB with the two currently available datasets, PTB [41] and STC [45]. CDTB is the only dataset that contains sequences captured with several devices in indoor and outdoor tracking scenes. STC [45] does in fact contain a few outdoor sequences, but these are confined to scenes without direct sunlight due to infra-red-based depth acquisition. The number of attributes is comparable to STC and much higher than PTB. The number of sequences (N_{seq}) is comparable to the currently largest dataset PTB, but CDTB exceeds the related datasets by an order of magnitude in the number of frames (N_{frm}). In fact, the average sequence of CDTB is approximately six times longer than in related datasets (N_{avg}), which affords a more accurate evaluation of long-term tracking properties.

A crucial tracker property required in many practical applications is target absence detection and target re-detection. STC lacks these events. The number of target disappearances followed by re-appearance in CDTB is comparable to PTB, but the disappearance periods (N_{out}) are much longer in CDTB. The average period of target absent (N_{avgout}) in PTB is approximately 6 frames, which means that only short-term occlusions are present. The average period of target absent in CDTB is nearly ten times larger, which allows tracker evaluation under much more challenging and realistic conditions.

Pose changes are much more frequent in CDTB than in the other two datasets. For example, the target undergoes a 180 degree out-of-plane rotation less than once per sequence in PTB and STC (N_{seqrot}). Since CDTB captures more dynamic scenarios, the target undergoes such pose change nearly 5 times per sequence.

The level of appearance change, realism, disappearances and sequence lengths result in a much more challenging dataset that allows performance evaluation more similar to the real-world tracking environment than STC and PTB. To quantify this, we evaluated trackers Ca3dMS, CSR-D and OTR on the three datasets and averaged their results. The trackers were evaluated on STC and CDTB using the PTB performance measure, since PTB does not provide ground truth bounding boxes for public evaluation.

Table 1 shows that the trackers achieve the highest performance on PTB, making it least challenging. The performance drops on STC, which supports the challenging small dataset diversity paradigm promoted in [45]. The performance further significantly drops on CDTB, which confirms that this dataset is the most challenging among the three.

5.3. Overall Tracking Performance

Figure 2 shows trackers ranked according to the F-measure, while tracking Precision-Recall plots are visualized for additional insights. A striking result is that the overall top-performing trackers are pure RGB trackers, which do not use depth information at all. MDNet and MBMD achieve comparable F-score, while FuCoLoT ranks third. It is worth mentioning that all three trackers are long-term with strong re-detection capability [33]. Even though MDNet was originally published as a short-term tracker, it has been shown that it performs well in a long-term scenario [33, 35, 43] due to its powerful CNN-based classi-

[4]KCF-D is modified by using depth as a feature channel in a correlation filter.

Table 1. Comparison of CDTB with related benchmarks in the number of RGB-D devices used for acquisition (N_{HW}), presence of indoor and outdoor sequences (In/Out), per-frame attribute annotation (Per-frame), number of attributes (N_{atr}), number of sequences (N_{seq}), total number of frames (N_{frm}) average sequence length (N_{avg}), number of frames with target not visible (N_{out}), number of target disappearances (N_{dis}), average length of target absence period (N_{avgout}), number of times a target rotates away from the camera by at least 180°(N_{rot}), average number of target rotations per sequence (N_{seqrot}) and tracking performance under the PTB protocol ($\Omega_{0.5}$).

Dataset	N_{HW}	In	Out	Per-frame	N_{atr}	N_{seq}	N_{frm}	N_{avg}	N_{out}	N_{avgout}	N_{dis}	N_{rot}	N_{seqrot}	$\Omega_{0.5}$
CDTB	3	✓	✓	✓	13	80	101,956	1,274	10,656	56.4	189	358	4.5	0.316
STC [45]	1	✓	✓	✓	12	36	9,195	255	0	0	0	30	0.8	0.530
PTB [41]	1	✓	✗	✗	5	95	20,332	214	846	6.3	134	83	0.9	0.749

fier with selective update and hard negative mining. Another long-term tracker, TLD, is ranked very low despite its re-detection capability, due to a fairly simplistic visual model which is unable to capture complex target appearance changes.

State-of-the-art RGB-D trackers, OTR and CSRDCF-D, using only hand-crafted features, achieve a comparable performance to complex deep-features-based short-term RGB trackers ECO and SiamFC. This implies that modern RGB deep features may compensate for the lack of depth information to some extent. On the other hand, state-of-the-art RGB trackers show improvements when extended by depth channel (CSRDCF-D, ECOhc-D and KCF-D). This means that existing RGB-D trackers lag behind the state-of-the-art RGB trackers which is a large opportunity for improvement by utilizing deep features combined with depth information.

Overall, both state-of-the-art RGB and RGB-D trackers exhibit a relatively low performance. For example, tracking Recall can be interpreted as the average overlap with ground truth on frames in which the target is visible. This value is below 0.5 for all trackers, which implies the dataset is particularly challenging for all trackers and offers significant potential for tracker improvement. Furthermore, we calculated tracking F-measure on sequences captured with each depth sensor. The results are comparable – 0.30 (ToF), 0.33 (Kinect) and 0.39 (stereo) – but they also imply that ToF is the most challenging and stereo is the least challenging sensor.

Precision-recall analysis. For further performance insights, we visualize the tracking Precision and Recall at the optimal tracking point, i.e., at the highest F-measure, in Figure 3. Precision and Recall are similarly low for most trackers, implying that trackers need to improve in target detection as well as localization accuracy. FuCoLoT, CSRDCF-D and TLD obtain significantly higher Precision than Recall, which means that mechanism for reporting loss of target is rather conservative in these trackers – a typical property we observed in all long-term trackers. The NCC tracker achieves significantly higher precision than recall, but this is a degenerated case since the target is reported as lost for most part of the sequence (very low Recall).

Another interesting observation is that tracking precision of the FuCoLoT is comparable to the top-performing MDNet and MBMD which shows that predictions made by FuColoT are similarly accurate to those made by top-performing trackers. On the other hand, top-performing MDNet and MBMD have a much higher recall, which shows that they are able to correctly track much more

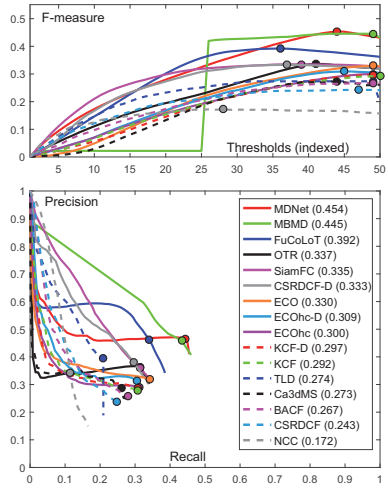

Figure 2. The overall tracking performance is presented as tracking F-measure (top) and tracking Precision-Recall (bottom). Trackers are ranked by their optimal tracking performance (maximum F-measure).

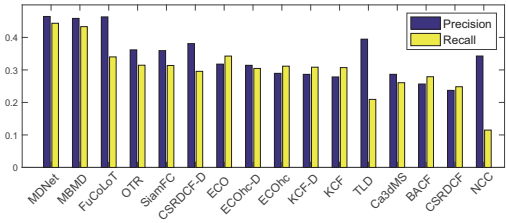

Figure 3. Tracking precision and recall calculated at the optimal point (maximum F-measure).

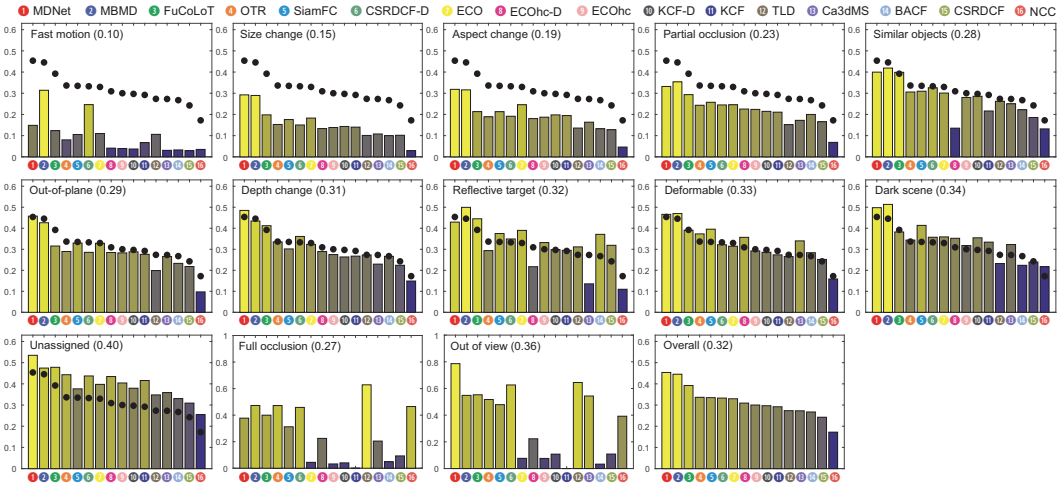

Figure 4. Tracking performance w.r.t. visual attributes. The first eleven attributes correspond to scenarios with a visible target (showing F-measure). The overall tracking performance is shown in each graph with black dots. The attributes *full occlusion* and *out of view* represent periods when the target is not visible and true negative rate is used to measure the performance.

frames where the target is visible, which might again be attributed to the use of deep features.

Overall findings. We can identify several good practices in the tracking architectures that look promising according to the overall results. Methods based on *deep features* show promise in capturing complex target appearance changes. We believe that *training* deep features on depth offers an opportunity for performance boost. A reliable *failure detection mechanism* is an important property for RGB-D tracking. Depth offers a convenient cue for detection of such events and combined with image-wide *re-detection* some of the RGB-D trackers address the long-term tracking scenario well. Finally, we believe that depth offers a rich *information complementary* to RGB for 3D target appearance modeling and depth-based target separation from the background, which can contribute in target localization. None of the existing RGB-D trackers incorporates all of these architectural elements, which opens a lot of new research opportunities.

5.4. Per-attribute Tracking Performance

The trackers were also evaluated on thirteen visual attributes (Section 3.2) in Figure 4. Performance on the attributes with visible target is quantified by the average F-measure, while true-negative rate (TNR [43]) is used to quantify the performance under full occlusion and out-of-view target disappearance.

Performance of all trackers is very low on *fast-motion*, making it the most challenging attribute. The reason for performance degradation is most likely the relatively small frame-to-frame target search range. Some of the long-term RGB-D and RGB trackers, e.g., MBMD and CSRDCF-D, stand out from the other trackers due to a well-designed image-wide re-detection mechanism, which compensates for a small frame-to-frame receptive field.

The next most challenging attributes are target *size change* and *aspect change*. MDNet and MBMD significantly outperform the other trackers since they explicitly estimate the target aspect. Size change is related to depth change, but the RGB-D tracker do not exploit this, which opens an opportunity for further research in depth-based robust scale adaptation.

Partial occlusion is particularly challenging for both RGB and RGB-D trackers. Failing to detect occlusion can lead to adaptation of the visual model to the occluding object and eventual tracking drift. In addition, too small frame-to-frame target search region leads to failure of target re-detection after the occlusion.

The attributes *similar objects*, *out-of-plane rotation*, *deformable*, *depth-change* and *dark scene* do not significantly degrade the performance compared to the overall performance. Nevertheless, the overall performance of trackers is rather low, which leaves plenty of room for improvements. We observe a particularly large drop in ECOhc-D on the similar-objects attribute which indicates that the tracker locks on to the incorrect/similar object at re-detection stage.

The *reflective target* attribute, unique for objects such as metal cups, mostly affects RGB-D trackers. The reason is that objects of this class are fairly well distinguished from the background in RGB, while their depth image is consistently unreliable. This means that more effort should be put

in information fusion part of the RGB-D trackers.

The attributes *deformable* and *dark-scene* are very well addressed by deep trackers (MDNet, MBMD, SiamFC and ECO), which makes them the most promising for coping with such situations. It seems that normalization, non-linearity and pooling in CNNs make deep features sufficiently invariant to image intensity changes and object deformations observed in practice.

Full occlusions are usually short-lasting events. On average, the trackers detect full a occlusion with some delay, thus a large percentage of occlusion frames are mistaken for the target visible. This implies poor ability to distinguish the appearance change due to occlusion from other appearance changes. The best target absence prediction at full occlusion is achieved by TLD, which is the most conservative in predicting target presence.

Situations when the target leaves the field of view (*out-of-view* attribute) are better predictable than full occlusions, due to longer target absence periods. Long-term trackers are performing very well in these situations and conservative visual model update seems to be beneficial.

A no-redetection experiment from [33] was performed to measure target re-detection capability in the considered trackers (Figure 5). In this experiment the standard tracking Recall (Re) is compared to a recall (Re_0) computed on modified tracker output – all overlaps are set to zero after the first occurrence of the zero overlap (i.e., the first target loss). Large difference between the recalls ($Re - Re_0$) indicates a good re-detection capability of a tracker. The trackers with the largest re-detection capability are MBMD, FuCoLoT (RGB trackers) and CSRDCF-D (RGB-D extension of CSRDCF) followed by OTR (RGB-D tracker) and two RGB trackers MDNet and SiamFc.

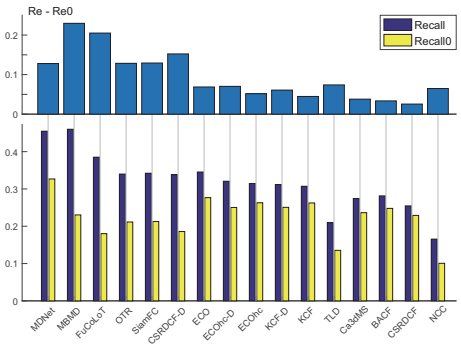

Figure 5. No redetection experiment. Tracking recall is shown on the bottom graph as dark blue bars. Modified tracking recall ($Re0$) is shown as yellow bars and it is calculated by setting the per-frame overlaps to zero after the first tracking failure. The difference between both recalls is shown on top. A large difference indicates good re-detection capability of the tracker.

6. Conclusion

We proposed a color-and-depth general visual object tracking benchmark (CDTB) that goes beyond the existing benchmarks in several ways. CDTB is the only benchmark with RGB-D dataset recorded by several color-and-depth sensors, which allows inclusion of indoor and outdoor sequences captured under unconstrained conditions (e.g., direct sun light) and covers a wide range of challenging depth signals. Empirical comparison to related datasets shows that CDTB contains a much higher level of object pose change and exceeds the other datasets in the number of frames by an order of magnitude. The objects disappear and reappear far more often, with disappearance periods ten times longer than in other benchmarks. Performance of trackers is lower on CDTB than related datasets. CDTB is thus currently the most challenging dataset, which allows RGB-D general object tracking evaluation under various realistic conditions involving target disappearance and re-appearance.

We evaluated recent state-of-the-art (SotA) RGB-D and RGB trackers on CDTB. Results show that SotA RGB trackers outperform SotA RGB-D trackers, which means that the architectures of RGB-D trackers could benefit from adopting (and adapting) elements of the recent RGB SotA. Nevertheless, the performance of all RGB and RGB-D trackers is rather low, leaving a significant room for improvements.

Detailed performance analysis showed several insights. Performance of baseline RGB trackers improved already from straightforward addition of the depth information. Current mechanisms for color and depth fusion in RGB-D trackers are inefficient and perhaps deep features trained on RGB-D data should be considered. RGB-D trackers do not fully exploit the depth information for robust object scale estimation. Fast motion is particularly challenging for all trackers indicating that short-term target search ranges should be increased. Target detection and mechanisms for detecting target loss have to be improved as well.

We believe these insights in combination with the presented benchmark will spark further advancements in RGB-D tracking and contribute to closing the gap between RGB and RGB-D state-of-the-art. Since the CDTB is a testing-only dataset we will work on constructing a large 6DOF dataset which could be used for training deep models for RGB-D tracking in the future.

Acknowledgements. This work is supported by Business Finland under Grant 1848/31/2015 and Slovenian research agency program P2-0214 and projects J2-8175 and J2-9433. J. Matas is supported by the Technology Agency of the Czech Republic project TE01020415 – V3C Visual Computing Competence Center.

References

[1] Ning An, Xiao-Guang Zhao, and Zeng-Guang Hou. Online RGB-D Tracking via Detection-Learning-Segmentation. In *ICPR*, 2016. 3

[2] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-Convolutional Siamese Networks for Object Tracking. In *ECCV Workshops*, 2016. 5

[3] Adel Bibi, Tianzhu Zhang, and Bernard Ghanem. 3D Part-Based Sparse Tracker with Automatic Synchronization and Registration. In *CVPR*, 2016. 2, 3

[4] David S. Bolme, J.Ross Beveridge, Bruce A. Draper, and Yui-Man Lui. Visual Object Tracking using Adaptive Correlation Filters. In *CVPR*, 2010. 3

[5] Anders Glent Buch, Dirk Kraft, Joni-Kristian Kamarainen, Henrik Gordon Petersen, and Norbert Krüger. Pose estimation using local structure-specific shape and appearance context. In *ICRA*, 2013. 2

[6] Massimo Camplani, Sion Hannuna, Majid Mirmehdi, Dima Damen, Adeline Paiement, Lili Tao, and Tilo Burghardt. Real-time RGB-D Tracking with Depth Scaling Kernelised Correlation Filters and Occlusion Handling. In *BMVC*, 2015. 3

[7] Changhyun Choi and Henrik Iskov Christensen. RGB-D object tracking: A particle filter approach on GPU. In *IROS*, 2013. 2

[8] Wongun Choi, Caroline Pantofaru, and Silvio Savarese. A General Framework for Tracking Multiple People from a Moving Camera. *IEEE PAMI*, 2013. 2

[9] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005. 3

[10] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient Convolution Operators for Tracking. In *CVPR*, 2017. 3, 5

[11] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc van Gool. A Mobile Vision System for Robust Multi-Person Tracking. In *CVPR*, 2008. 2

[12] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Correlation Filters with Limited Boundaries. In *CVPR*, 2015. 3

[13] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In *CVPR*, 2018. 2

[14] Sion Hannuna, Massimo Camplani, Jake Hall, Majid Mirmehdi, Dima Damen, Tilo Burghardt, Adeline Paiement, and Lili Tao. DS-KCF: A Real-time Tracker for RGB-D Data. *Journal of Real-Time Image Processing*, 2016. 3

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004. 4

[16] Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-Speed Tracking with Kernelized Correlation Filters. *IEEE PAMI*, 37(3):583–596, 2015. 3, 5

[17] Heiko Hirschmuller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In *CVPR*, 2005. 3

[18] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-Learning-Detection. *IEEE PAMI*, 34(7):1409–1422, 2011. 3, 5

[19] Ugur Kart, Joni-Kristian Kämäräinen, and Jiri Matas. How to Make an RGBD Tracker ? In *ECCV Workshops*, 2018. 3, 5

[20] Ugur Kart, Joni-Kristian Kämäräinen, Jiri Matas, Lixin Fan, and Francesco Cricri. Depth Masked Discriminative Correlation Filter. In *ICPR*, 2018. 3

[21] Ugur Kart, Alan Lukežič, Matej Kristan, J.-K. Kämäräinen, and J. Matas. Object Tracking by Reconstruction with View-Specific Discriminative Correlation Filters. In *CVPR*, 2019. 3, 5

[22] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning Background-Aware Correlation Filters for Visual Tracking. In *ICCV*, 2017. 5

[23] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomas Vojír, and et al. The Visual Object Tracking VOT2016 Challenge Results. In *ECCV Workshops*, 2016. 1

[24] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, and et al. The Visual Object Tracking VOT2017 Challenge Results. In *ICCV Workshops*, 2017. 1

[25] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, and Tomas Vojir et al. The sixth Visual Object Tracking VOT2018 challenge results. In *ECCV Workshops*, 2018. 1, 3, 4

[26] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, and Luka et al. Čehovin Zajc. The Visual Object Tracking VOT2015 Challenge Results. In *ICCV Workshops*, 2015. 1, 3

[27] Matej Kristan, Jiri Matas, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE PAMI*, 38(11):2137–2155, 2016. 1

[28] Matej Kristan, Roman Pflugfelder, Ales Leonardis, Jiri Matas, Luka Čehovin, Georg Nebehay, Tomas Vojír, and et al. The Visual Object Tracking VOT2014 Challenge Results. In *ECCV Workshops*, 2014. 1, 4

[29] Matej Kristan, Roman Pflugfelder, Ales Leonardis, Jiri Matas, Fatih Porikli, and et al. The Visual Object Tracking VOT2013 Challenge Results. In *CVPR Workshops*, 2013. 5

[30] Ye Liu, Xiao-Yuan Jing, Jianhui Nie, Hao Gao, Jun Liu, and Guo-Ping Jiang. Context-aware 3-D Mean-shift with Occlusion Handling for Robust Object Tracking in RGB-D Videos. *IEEE TMM*, 2018. 3, 5

[31] Alan Lukežič, Luka Čehovin Zajc, Tom'aš Vojíř, Jiř'i Matas, and Matej Kristan. FuCoLoT - A Fully-Correlational Long-Term Tracker. In *ACCV*, 2018. 3, 5

[32] Alan Lukežič, Tomas Vojír, Luka Čehovin, Jiri Matas, and Matej Kristan. Discriminative Correlation Filter with Channel and Spatial Reliability. In *CVPR*, 2017. 3, 5

[33] Alan Lukezic, Luka Cehovin Zajc, Tomás Vojír, Jiri Matas, and Matej Kristan. Now you see me: evaluating performance in long-term visual tracking. *CoRR*, abs/1804.07056, 2018. 2, 4, 5, 8

[34] Kourosh Meshgi, Shin ichi Maeda, Shigeyuki Oba, Henrik Skibbe, Yu zhe Li, and Shin Ishii. An Occlusion-aware Particle Filter Tracker to Handle Complex and Persistent Occlusions. *CVIU*, 150:81 – 94, 2016. 3

[35] Abhinav Moudgil and Vineet Gandhi. Long-Term Visual Object Tracking Benchmark. In *ACCV*, 2018. 5

[36] Matthias Mueller, Neil Smith, and Bernard Ghanem. A Benchmark and Simulator for UAV Tracking. In *ECCV*, 2016. 1

[37] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In *ECCV*, 2018. 1

[38] Hyeonseob Nam and Bohyung Han. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In *CVPR*, 2016. 3, 5

[39] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. In *ECCV*, 2016. 2

[40] Arnold W. M. Smeulders, Dung Manh Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual Tracking: An Experimental Survey. *IEEE PAMI*, 36(7):1442–1468, 2014. 1

[41] Shuran Song and Jianxiong Xiao. Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. In *ICCV*, 2013. 2, 3, 5, 6

[42] Luciano Spinello and Kai Oliver Arras. People detection in RGB-D data. In *IROS*, 2011. 2

[43] Jack Valmadre, Luca Bertinetto, João F. Henriques, Ran Tao, Andrea Vedaldi, Arnold W. M. Smeulders, Philip H. S. Torr, and Efstratios Gavves. Long-term Tracking in the Wild: A Benchmark. In *ECCV*, 2018. 2, 5, 7

[44] Yi Wu, Jongwoo Lim, and Yang Ming-Hsuan. Object Tracking Benchmark. *IEEE PAMI*, 37:1834 – 1848, 2015. 1

[45] Jingjing Xiao, Rustam Stolkin, Yuqing Gao, and Ales Leonardis. Robust Fusion of Color and Depth Data for RGB-D Target Tracking Using Adaptive Range-Invariant Depth Models and Spatio-Temporal Consistency Constraints. *IEEE Transactions on Cybernetics*, 48:2485 – 2499, 2018. 2, 3, 5, 6

[46] Yunhua Zhang, Dong Wang, Lijun Wang, Jinqing Qi, and Huchuan Lu. Learning Regression and Verification Networks for Long-term Visual Tracking. *CoRR*, abs/1809.04320, 2018. 3, 5