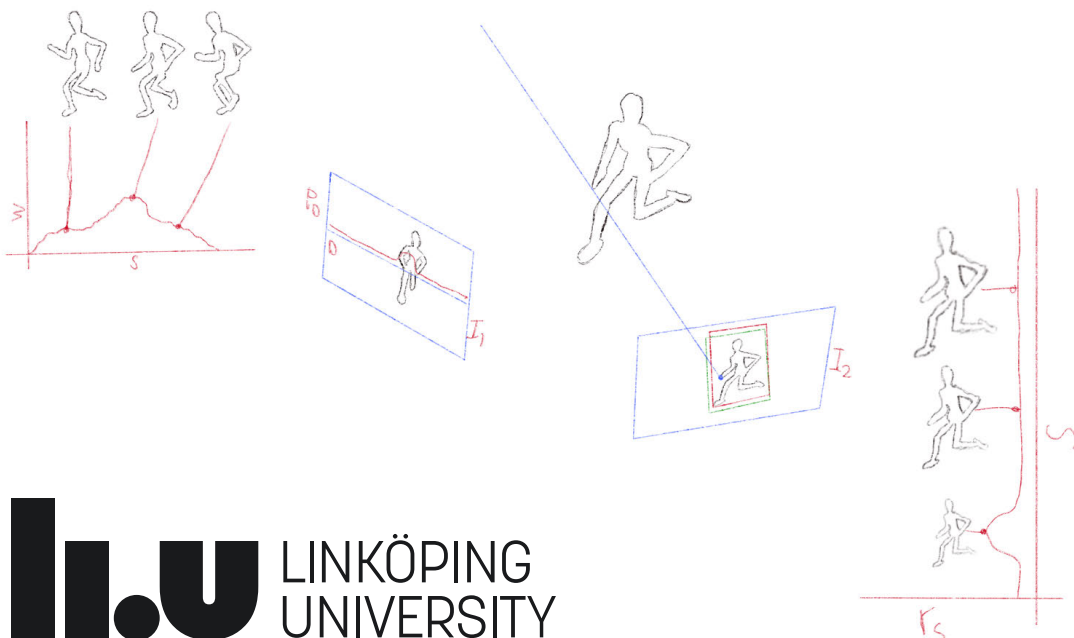


Learning Visual Perception for Autonomous Systems

Gustav Häger



Linköping Studies in Science and Technology
Dissertations, No. 2138

Learning Visual Perception for Autonomous Systems

Gustav Häger



Linköping University
Department Of Electrical Engineering
Computer Vision Laboratory
SE-581 83 Linköping, Sweden

Linköping 2021



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

<https://creativecommons.org/licenses/by-nc/4.0/>

Edition 1:1

© Gustav Häger, 2021

ISBN 978-91-7929-671-1

ISSN 0345-7524

URL <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-175177>

Published articles have been reprinted with permission from the respective copyright holder.

Typeset using X_YT_EX

Printed by LiU-Tryck, Linköping 2021

POPULÄRVETENSKAPLIG SAMMANFATTNING

De senaste årens allt snabbare utveckling av beräkningshårdvara, sensorer och mjukvarutekniker har gjort det möjligt att skapa allt mer autonoma system. Sådana kan variera i autonomigrad från ett antisladdsystem för en i övrigt manuellt kontrollerad bil, till system för kollisionsundvikning i en manuellt kontrollerad drönare, till en helt autonom bil eller annan farkost. Med en ökande förmåga att arbeta självständigt utan mänsklig övervakning ökar också bredden på möjliga situationer som systemen förväntas hantera.

Gemensamt för många, om inte alla, autonoma system är att de behöver en korrekt och updaterad bild av sin omgivning för att kunna agera på ett intelligent sätt. En lång rad av sensorer som gör detta möjligt finns tillgängliga, där kameror är en av de mest mångsidiga. Jämfört med andra typer av sensorer har kameror en rad fördelar, som att de är relativt billiga, passiva, och kan användas utan krav på extern infrastruktur. Det visuella data som kameror genererar kan användas för att följa externa objekt, bestämma positionen för kameran själv, eller beräkna avstånd.

Att framgångsrikt utnyttja möjligheterna i denna information kräver dock att en lång rad tekniska problem hanteras. Många av dessa problem är grundar sig i att kunna känna igen att två bildregioner från olika tidpunkter eller betraktningvinklar avbildar samma sak.

Ett typexempel på ett sådant problem är det visuella följningsproblemet. I det visuella följningsproblemet är målet att bestämma ett objekts position och storlek för alla bilder i en sekvens av bilder. I allmänhet är objektets utseende inte känt av algoritmen, utan en utseendemodell måste skapas succesivt med hjälp av maskininlärning.

Problem som liknar detta förekommer inom många andra områden av datorseende, speciellt inom geometri. Inom många geometriska problem krävs det till exempel att man finner korresponderande punkter i ett flertal bilder.

Den första samlingen av bidrag i denna avhandling behandlar det visuella följningsproblemet. De föreslagna metoderna är baserade på en adaptiv utseendemodell kallad diskriminativa korrelationsfilter. I det första bidraget till sådana metoder utökas ramverket till att skatta ett objekts storlek såväl som position. Ett andra bidrag undersöker hur korrelationsfilterbaserade metoder kan utökas till att även utnyttja visuella särdrag som har framställt med hjälp av maskininlärning.

En andra samling med bidrag behandlar utvärdering av metoder för visuell följning. Dels inom den årligt förekommande tävlingen visual object tracking challenge. Ett andra bidrag till utvärderingsmetodig inom visuell följning syftar till att undvika fallgropar som lätt uppkommer då metoder anpassas allt för väl för måtten som används för att utvärdera dem.

En tredje samling med bidrag relaterar till olika sätt att hantera situationer då inlärningsprocessen i de tidigare beskrivna följningsmetoderna introducerar felaktiga data i modellen. Detta görs i ett första bidrag i ett robotiksystem för följning av människor i en ostrukturerad miljö. Ett andra bidrag är baserat på dynamisk omviktning av tidigare samlad data för att dynamiskt vikta ned datapunkter som inte representerar det följda objektet väl. I ett sista bidrag undersöks hur en prediktions osäkerhet kan skattas samtidigt som prediktionen själv.

ABSTRACT

In the last decade, developments in hardware, sensors and software have made it possible to create increasingly autonomous systems. These systems can be as simple as limited driver assistance software lane-following in cars, or limited collision warning systems for otherwise manually piloted drones. On the other end of the spectrum exist fully autonomous cars, boats or helicopters. With increasing abilities to function autonomously, the demands to operate with minimal human supervision in unstructured environments increase accordingly.

Common to most, if not all, autonomous systems is that they require an accurate model of the surrounding world. While there is currently a large number of possible sensors useful to create such models available, cameras are one of the most versatile. From a sensing perspective cameras have several advantages over other sensors in that they require no external infrastructure, are relatively cheap and can be used to extract such information as the relative positions of other objects, their movements over time, create accurate maps and locate the autonomous system within these maps.

Using cameras to produce a model of the surroundings require solving a number of technical problems. Often these problems have a basis in recognizing that an object or region of interest is the same over time or in novel viewpoints. In visual tracking this type of recognition is required to follow an object of interest through a sequence of images. In geometric problems it is often a requirement to recognize corresponding image regions in order to perform 3D reconstruction or localization.

The first set of contributions in this thesis is related to the improvement of a class of on-line learned visual object trackers based on discriminative correlation filters. In visual tracking estimation of the objects size is important for reliable tracking, the first contribution in this part of the thesis investigates this problem. The performance of discriminative correlation filters is highly dependent on what feature representation is used by the filter. The second tracking contribution investigates the performance impact of different features derived from a deep neural network.

A second set of contributions relate to the evaluation of visual object trackers. The first of these are the visual object tracking challenge. This challenge is a yearly comparison of state-of-the art visual tracking algorithms. A second contribution is an investigation into the possible issues when using bounding-box representations for ground-truth data.

In real world settings tracking typically occur over longer time sequences than is common in benchmarking datasets. In such settings it is common that the model updates of many tracking algorithms cause the tracker to fail silently. For this reason it is important to have an estimate of the trackers performance even in cases when no ground-truth annotations exist. The first of the final three contributions investigates this problem in a robotics setting, by fusing information from a pre-trained object detector in a state-estimation framework. An additional contribution describes how to dynamically re-weight the data used for the appearance model of a tracker. A final contribution investigates how to obtain an estimate of how certain detections are in a setting where geometrical limitations can be imposed on the search region. The proposed solution learns to accurately predict stereo disparities along with accurate assessments of each predictions certainty.

Författarens tack

Den här avhandlingen hade inte blivit av utan en del andra personer.

Mattias, som fick mig att ens börja med det här.

Martin, som såg till att en det gick rätt bra, speciellt i början.

Fahad, som visade mig hur man skriver så att andra orkar läsa.

Michael, som jag haft många mer eller mindre givande diskussioner med.

Evelina, som har varit bra fikasällskap under tiden.

Mina föräldrar, som gärna får inse att har varit vuxen ett tag nu.

Emil, som kommer få hjälpa mig att laga mat till festen, någon gång.

Ellen och George, som lät mig använda Cosmo för många exempelbilder.

Diverse personer från IDA, ISY, WASP, LiTheBlås och spexet.

Säkert många fler, som jag bara inte kommit ihåg att skriva om här.

Anställda vid CVL, tidigare och nuvarande. Speciellt: Andreas, Abdo, Felix och Mikael som hjälpte mig korrekturläsa avhandlingen.

Contents

Abstract	iii
Acknowledgments	v
Contents	vii
I Background	1
1 Introduction	3
1.1 Introduction	3
1.2 Contributions	5
1.3 Thesis outline	6
1.4 Included publications	7
1.5 Additional publications	14
2 Visual Object tracking	15
2.1 Correlation filter trackers	16
2.2 Extending the MOSSE tracker	19
3 Evaluation of visual object trackers	25
3.1 The visual object tracking challenge	25
3.2 Issues with using intersection over union scores	28
4 Tracking in practice and uncertainty	31
4.1 Tracking in practice	32
4.2 Retroactively weighting samples	34
4.3 Recognizing image regions with uncertainty	35
5 Concluding remarks	41
Bibliography	43

II	Publications	49
A	Discriminative scale space tracking	51
B	Convolutional features for correlation filter based visual tracking	69
C	The visual object tracking VOT 2017 challenge results	81
D	Countering bias in tracking evaluations	109
E	Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking	119
F	Combining visual tracking and person detection for long term tracking on a uav	131
G	Predicting disparity distributions	147

PART I

BACKGROUND

INTRODUCTION

1.1 Introduction

For most living creatures, using visual perception for recognition and navigation is as natural as breathing. In most cases, no active effort is needed to separate objects of interest from the background, even as they change position in the visual field. In the same way, we can without conscious effort recognize that two separate images depict the same thing, even when they have drastically different viewpoints.

Designing an artificial vision system with the same capabilities as those of humans or animals remains unsolved despite early optimism¹. Early attempts at creating artificial vision systems, such as the one used by the robot Shakey [50] relied on controlled environments to function. This is a sharp contrast to biological vision that works equally well in most situations. Unlike the early attempts at artificial vision, biological vision appears to be at least partly a learned ability.

For a newborn human the visual system develops gradually over the first few months of life [1]. Other species can learn complex skills such as stabilization of flight comparatively quickly, yet still require a period of learning to navigate their local environment [4]. Early attempts at integrating learning into vision systems showed some success for following real-world roads [45] by learning the road appearance from image pixels using an artificial neural network.

Modern autonomous systems have come a long way from these early attempts, and it is now routine with vision-based lane-following systems in cars, or collision avoidance based on image data in many settings. A large part of this is due to the increase in computational power available, making it possible

¹For example the attempt to solve object detection in one summer <http://hdl.handle.net/1721.1/6125>

to utilize machine-learning techniques. This in turn leads to more adaptive systems, that can function well in a wider range of situations.

Creating fully autonomous systems capable of operation entirely without human supervision remains an open research problem. An ongoing research program in this area is the Wallenberg AI, Autonomous systems and Software program (WASP). The WASP program is a large-scale research effort aimed at increasing the general level of competence in the areas of AI and autonomous systems in Sweden. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

As a part of the WASP program a number of research arenas have been defined relating to various aspects of autonomous systems. The work in this thesis is most closely connected to the WARA-PS arena. The WARA-PS arena aims to simulate a rescue scenario in a maritime setting, with a large number of autonomous and manually controlled boats and helicopters involved in finding distressed persons.

Most of the work presented in this thesis concerns different tasks of visual recognition relevant to autonomous systems, primarily in the domain of visual object tracking. Visual object tracking is the task of tracking an object through sequence of images. The approach for solving this problem taken in this thesis is to formulate this as a problem of recognizing, or detecting the object of interest in all images of the sequence. The methods presented here utilize a learned appearance model that is updated with additional data collected while tracking.

This appearance model will sometimes fail to detect the tracked object. This happens in situations such as when the tracked objects apparent size in the image changes to a significant degree. Successfully handling scale-variations can be done by explicitly modeling this in the tracking algorithm. The first contribution in this thesis deals with scale-estimation for visual tracking. Other situations such as motion blur or deformations of the tracked object remain challenging. For dealing with such situations the perhaps most important component of a tracker is the feature representation used. A second contribution in this thesis utilizes powerful learned feature representations to improve tracking performance in a wide range of situations.

In order to have a good estimate of a tracking algorithms performance on real data it is insufficient to evaluate on only a small number of sequences. For this reason large-scale benchmarks for evaluation of trackers have been collected. One such benchmark is the visual-object-tracking challenge (VOT) [32]. The evaluation metrics in benchmarks such as VOT are based on comparing the output of a tracker with human made annotations. These comparisons are reliant on metrics comparing how well image regions correspond to one-another. One contribution of this thesis is an investigation into the behavior of these metrics in some extreme situations where objects cover the majority of the image.

However the evaluation metrics used in such benchmarks are not guaranteed to provide reliable assessments of the evaluated trackers performance in all situations. One contribution of this thesis is an investigation into the behavior of the intersection over union measure that is often used to evaluate tracking algorithms.

In order to utilize tracking in real-world scenarios additional challenges occur, often as a result of the on-line learning utilized by the appearance models. This is particularly common in the case of long-term tracking, where minor errors in tracking will compound over time, leading to drift in the visual appearance model. Model drift is addressed in two ways in this thesis. The first approach utilizes a combination of state-space modeling and an object detector with a fixed model to detect and recover from model drift. The second approach retroactively re-weights collected samples.

Ideally however, it is preferable if the output of a detection included an assessment of its reliability. One way of producing uncertainties from predictions is to predict a distribution of possible detections instead of a single point-estimate. This is investigated in the final contribution of this thesis in the context of a stereo disparity estimation based on an end-to-end learned neural network.

1.2 Contributions

The majority of the contributions in this thesis related in some way to visual object tracking. The first part of this thesis discusses this problem, with possible solutions based on discriminative correlation filters (DCF). These trackers form a family of adaptive on-line learned appearance models based on application of a linear classifier.

A second set of contributions relate to the evaluation of visual object trackers. The author was involved in the technical committee of the visual object tracking challenge for the years 2015, 2016, 2017. A further contribution to the evaluation of visual object trackers is an investigation into problematic situations that occur when using an axis-aligned bounding-box target representation.

A third set of contributions relate to the difficulties that occur in many real-world tracking situations. In such settings tracking must often be done over much longer term than is typical in benchmark situations. This requires careful management of the learned components of a tracking algorithm. Several approaches for this are proposed. The first is based on utilizing an explicit recovery mechanism, in combination with an object detector. A second approach utilizing successive re-weighting of training data. A final method simplify the more general recognition problem such that it is possible to explicitly produce a distribution over the possible locations of a sought after image region.

1.3 Thesis outline

The first part of this thesis describes the visual-tracking problem, and introduces the discriminative correlation filter based framework. The second part discusses evaluation of these trackers and issues with using bounding boxes representations for ground-truth and tracker output. The final part investigates a practical system for following humans using a quadcopter as well as a method for obtaining uncertainties from a deep-learning based stereo algorithm.

1.4 Included publications

A: “Discriminative scale space tracking”

M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. “Discriminative scale space tracking.” In: *IEEE transactions on pattern analysis and machine intelligence* 39.8 (2016), pp. 1561–1575

Abstract: Accurate scale estimation of a target is a challenging research problem in visual object tracking. Most state-of-the-art methods employ an exhaustive scale search to estimate the target size. The exhaustive search strategy is computationally expensive and struggles when encountered with large scale variations. This paper investigates the problem of accurate and robust scale estimation in a tracking-by-detection framework. We propose a novel scale adaptive tracking approach by learning separate discriminative correlation filters for translation and scale estimation. The explicit scale filter is learned online using the target appearance sampled at a set of different scales. Contrary to standard approaches, our method directly learns the appearance change induced by variations in the target scale. Additionally, we investigate strategies to reduce the computational cost of our approach. Extensive experiments are performed on the OTB and the VOT2014 datasets. Compared to the standard exhaustive scale search, our approach achieves a gain of 2.5% in average overlap precision on the OTB dataset. Additionally, our method is computationally efficient, operating at a 50% higher frame rate compared to the exhaustive scale search. Our method obtains the top rank in performance by outperforming 19 state-of-the-art trackers on OTB and 37 state-of-the-art trackers on VOT2014.

Contributions: The initial idea was from Martin Danelljan, the author wrote the implementation, ran the experiments and contributed to the manuscript.

B: “Convolutional features for correlation filter based visual tracking”

M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg.
“Convolutional features for correlation filter based visual tracking.” In: *Proceedings of the IEEE international conference on computer vision workshops*. 2015, pp. 58–66

Abstract: Visual object tracking is a challenging computer vision problem with numerous real-world applications. This paper investigates the impact of convolutional features for the visual tracking problem. We propose to use activations from the convolutional layer of a CNN in discriminative correlation filter based tracking frameworks. These activations have several advantages compared to the standard deep features (fully connected layers). Firstly, they mitigate the need of task specific fine-tuning. Secondly, they contain structural information crucial for the tracking problem. Lastly, these activations have low dimensionality. We perform comprehensive experiments on three benchmark datasets: OTB, ALOV300++ and the recently introduced VOT2015. Surprisingly, different to image classification, our results suggest that activations from the first layer provide superior tracking performance compared to the deeper layers. Our results further show that the convolutional features provide improved results compared to standard hand-crafted features. Finally, results comparable to state-of-the-art trackers are obtained on all three benchmark datasets.

Contributions: The author and Martin Danelljan contributed equally to manuscript, idea and implementation.

C: “The visual object tracking VOT 2017 challenge results”

M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin Zajc, T. Vojir, G. Häger, A. Lukežič, A. Eldesokey, et al. “The visual object tracking VOT 2017 challenge results.” In: *Proceedings of the IEEE international conference on computer vision workshops*. 2017, pp. 1949–1972

Abstract: The Visual Object Tracking challenge VOT2017 is the fifth annual tracker benchmarking activity organized by the VOT initiative. Results of 51 trackers are presented; many are state-of-the-art published at major computer vision conferences or journals in recent years. The evaluation included the standard VOT and other popular methodologies and a new “real-time” experiment simulating a situation where a tracker processes images as if provided by a continuously running sensor. Performance of the tested trackers typically by far exceeds standard baselines. The source code for most of the trackers is publicly available from the VOT page. The VOT2017 goes beyond its predecessors by (i) improving the VOT public dataset and introducing a separate VOT2017 sequestered dataset, (ii) introducing a realtime tracking experiment and (iii) releasing a redesigned toolkit that supports complex experiments. The dataset, the evaluation kit and the results are publicly available at the challenge website.

Contributions: The author contributed with evaluating submitted methods on the held-out dataset, and contributed to the final manuscript.

D: “Countering bias in tracking evaluations”

G. Häger, M. Felsberg, and F. S. Khan. “Countering bias in tracking evaluations.” In: *13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, January 27-29, Funchal, Madeira*. Vol. 5. Science and Technology Publications, Lda. 2018, pp. 581–587

Abstract: Recent years have witnessed a significant leap in visual object tracking performance mainly due to powerful features, sophisticated learning methods and the introduction of benchmark datasets. Despite this significant improvement, the evaluation of state-of-the-art object trackers still relies on the classical intersection over union (IoU) score. In this work, we argue that the object tracking evaluations based on classical IoU score are sub-optimal. As our first contribution, we theoretically prove that the IoU score is biased in the case of large target objects and favors over-estimated target prediction sizes. As our second contribution, we propose a new score that is unbiased with respect to target prediction size. We systematically evaluate our proposed approach on benchmark tracking data with variations in relative target size. Our empirical results clearly suggest that the proposed score is unbiased in general.

Contributions: Michael Felsberg contributed the initial idea. The author designed and implemented the experiments and wrote the manuscript.

E: “Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking”

M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg.
“Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1430–1438

Abstract: Tracking-by-detection methods have demonstrated competitive performance in recent years. In these approaches, the tracking model heavily relies on the quality of the training set. Due to the limited amount of labeled training data, additional samples need to be extracted and labeled by the tracker itself. This often leads to the inclusion of corrupted training samples, due to occlusions, misalignments and other perturbations. Existing tracking-by-detection methods either ignore this problem, or employ a separate component for managing the training set. We propose a novel generic approach for alleviating the problem of corrupted training samples in tracking-by-detection frameworks. Our approach dynamically manages the training set by estimating the quality of the samples. Contrary to existing approaches, we propose a unified formulation by minimizing a single loss over both the target appearance model and the sample quality weights. The joint formulation enables corrupted samples to be down-weighted while increasing the impact of correct ones. Experiments are performed on three benchmarks: OTB-2015 with 100 videos, VOT2015 with 60 videos, and Temple-Color with 128 videos. On the OTB-2015, our unified formulation significantly improves the baseline, with a gain of 3.8% in mean overlap precision. Finally, our method achieves state-of-the-art results on all three datasets.

Contributions: The author implemented and ran experiments based on idea from Martin Danelljan, the author designed the forgetting factor and participated in writing the manuscript.

F: “Combining visual tracking and person detection for long term tracking on a uav”

G. Häger, G. Bhat, M. Danelljan, F. S. Khan, M. Felsberg, P. Rudl, and P. Doherty. “Combining visual tracking and person detection for long term tracking on a uav.” In: *International Symposium on Visual Computing*. Springer. 2016, pp. 557–568

Abstract: Visual object tracking performance has improved significantly in recent years. Most trackers are based on either of two paradigms: online learning of an appearance model or the use of a pretrained object detector. Methods based on online learning provide high accuracy, but are prone to model drift. The model drift occurs when the tracker fails to correctly estimate the tracked object’s position. Methods based on a detector on the other hand typically have good long-term robustness, but reduced accuracy compared to online methods. Despite the complementarity of the aforementioned approaches, the problem of fusing them into a single framework is largely unexplored. In this paper, we propose a novel fusion between an online tracker and a pretrained detector for tracking humans from a UAV. The system operates at real-time on a UAV platform. In addition we present a novel dataset for long-term tracking in a UAV setting, that includes scenarios that are typically not well represented in standard visual tracking datasets.

Contributions: The author implemented the tracking algorithm and designed the fusion framework. The author wrote the manuscript.

G: “Predicting disparity distributions”

G. Häger, M. Persson, and M. Felsberg. “Predicting disparity distributions.” In: *2021 International Conference on Robotics and Automation (ICRA) (in print)*. 2021

Abstract: We investigate a novel deep-learning-based approach to estimate uncertainty in stereo disparity prediction networks. Current state-of-the-art methods often formulate disparity prediction as a regression problem with a single scalar output in each pixel. This can be problematic in practical applications as in many cases there might not exist a single well defined disparity, for example in cases of occlusions or at depth-boundaries. While current neural-network-based disparity estimation approaches obtain good performance on benchmarks, the disparity prediction is treated as a black box at inference time. In this paper we show that by formulating the learning problem as a regression with a distribution target, we obtain a robust estimate of the uncertainty in each pixel, while maintaining the performance of the original method. The proposed method is evaluated both on a large-scale standard benchmark, as well on our own data. We also show that the uncertainty estimate significantly improves by maximizing the uncertainty in those pixels that have no well defined disparity during learning.

Contributions: The author contributed with the idea, implementation, experiments and wrote the manuscript.

1.5 Additional publications

This section lists peer-reviewed publications that were not included in the thesis.

M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. “Learning spatially regularized correlation filters for visual tracking.” In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4310–4318

This paper is the base for paper B as well as paper E. It extends the discriminative correlation filtering framework by extending the tracking framework to have spatially varying regularization.

M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. “Coloring channel representations for visual tracking.” In: *Scandinavian Conference on Image Analysis*. Springer. 2015, pp. 117–129

This paper investigates the impact of combining channel-coded representations along with the color-name representation in a discriminative correlation filter tracking framework.

VISUAL OBJECT TRACKING

In visual object tracking the goal is to track one or several object through a sequence of images, often without prior knowledge of the object type. In most cases the output is the position and size of the tracked object. The target-region in each image is represented either as a bounding-box or using more fine-grained representation such as bounding-polygons or per-pixel segmentations.

This chapter serves as an introduction to a class of visual object tracking algorithms that are capable of tracking a single-target of unknown type by processing a sequence of images causally. Single target track can be viewed as a simplified version of a more general tracking problem, where in the general setting it is also required to detect the objects to track, and the number of objects is not known. The single target problem is still useful, as improvements in single-target tracking easily translate to improvements in multi-target-tracking.

The visual object tracking setting has some significant differences when compared to object tracking in other sensor modalities. For sensors producing direct range measurements such as LiDAR or radar, association of measurements with tracked objects is commonly done using state space representations in combination with motion models of the tracked object. In image data, the tracked object is often more robustly represented in terms of its visual appearance.

In practice this means that when tracking in image data, approaches based on state-space models and motion priors, such as Kalman filters [29], tend to have less utility than when tracking using other types of sensors. Instead a model representing the appearance of a tracked object can be constructed directly from the image pixels, in order to distinguish image regions containing the tracked object from those that do not. That is tracking-by-detection is done by learning a target-specific appearance model for the object that is tracked.

Early visual tracking methods, such as the Kanade-Lucas-Tomasi tracker[40], formulated the problem of recognizing a tracked region in a new image as an optimization problem. In these methods a photometric error between an image patch and regions in a new image is minimized with respect to a transform between the images to find the patch position in the new images. Many early tracking methods did not utilize a more sophisticated model of the tracked region than the region itself.

Later methods attempted to utilize more sophisticated models for the tracked object. These methods are still often based on a form of template-matching, where the template is updated over time. As the template is updated with new information obtained during tracking, minor errors in position will often result in model drift[41], where the appearance model eventually represents something other than the intended region.

Dealing with model drift is difficult. In general the most straightforward solution is to improve the tracking such that the model updates are more correct. This can be done in ways that explicitly handle situations likely to cause model drift. Such situations include occlusions, changes in the shape of the object, distractor objects with similar appearance to the tracked object, or the gradual accumulation of tracking errors. Of these changes in the size of the tracked target is almost guaranteed to happen unless tracking is limited to objects at a fixed distance. Paper A investigates ways to address changes in scale. General tracking performance is often dependent on the feature representations used. Applying high-dimensional learned features extracted from a neural network to tracking problems is investigated in paper B.

2.1 Correlation filter trackers

The trackers in this thesis are based on the framework of discriminative correlation filters (DCF). These methods can in many ways be viewed as a discriminative counterpart to correlation based template matching. This introductory section will derive a simple baseline DCF tracker, called the MOSSE [3].

Starting with an image patch x_t , at time t , containing the region or object we wish to track we can for a region of the same size in a new image x_{t+1} calculate a score y for each position (r, c) as

$$y(r, c) = \sum_{k=0}^K \sum_{n=0}^N x_t(k, n) x_{t+1}(r+k, c+n). \quad (2.1)$$

Where N and K denote the size of the image patch. In order to reduce notational clutter this operation between x_t and x_{t+1} will be written as $x_t \star x_{t+1}$. If the corresponding standard deviation of x_t and x_{t+1} are assumed to be one and the mean zero, this is the same as the correlation between x_t and x_{t+1} .

In this thesis this normalization of subtracting the mean and scaling by variance is considered a pre-processing step. As images have a known numerical range, it is often more convenient to utilize this for normalization instead. When using more advanced feature representations some kind of normalization is often included in the feature itself. This means further operations of this type are either meaningless or possibly counter-productive. As a final remark, images and image patches are zero-padded unless otherwise noted.

Updating the position of the tracker can be done by taking the position of the maximum of the correlation scores y as the new position for the tracked object. This process can then be repeated for additional images, giving a position of the object in each input image.

In this framework the appearance model used is still the image patch x_t itself. The DCF framework improves on this by instead attempting to find a model that is as discriminative as possible when compared to the background. This model can be found by creating a fixed output shape y , and using this as labels in a machine-learning formulation.

In many classification and detection problems the training data consists of image patches that either contain the target or not. These training patches are often labeled as either positive or negative [23, 59]. Example patches that partially cover the tracked object are difficult to label using this strict positive-negative sample approach. Instead it is possible to label patches using a decaying positive label, where the patch that is correctly centered on the tracked object is labeled as entirely positive and shifted patches have increasingly negative labels.

This results in having a continuous labeling function, where the most-correctly-centered patch has label one, those that are offset by only a few pixels have only slightly lower label, and further offset samples decaying to a label value of zero. These labels can be assigned using a Gaussian function g , centered on the target. That is we can think the ideal output being g when the template from (2.1) is evaluated on a new image.

More formally, given an image patch x_t , we use our fix labeling function g to create the best possible template for detecting it in a new frame. This can be done by solving

$$\min_{h_t} \sum_{r,c} (x_t(r,c) \star h_t(r,c) - g(r,c))^2, \quad (2.2)$$

where g is typically having some small variance. Finding the position of the tracked object in a new image can be done by calculating the response as $y = h_t \star x_{t+1}$. Note that as before there are no particular constraints on the shape of y , though it is often similar to the desired output g , there is no guarantee that this is the case.

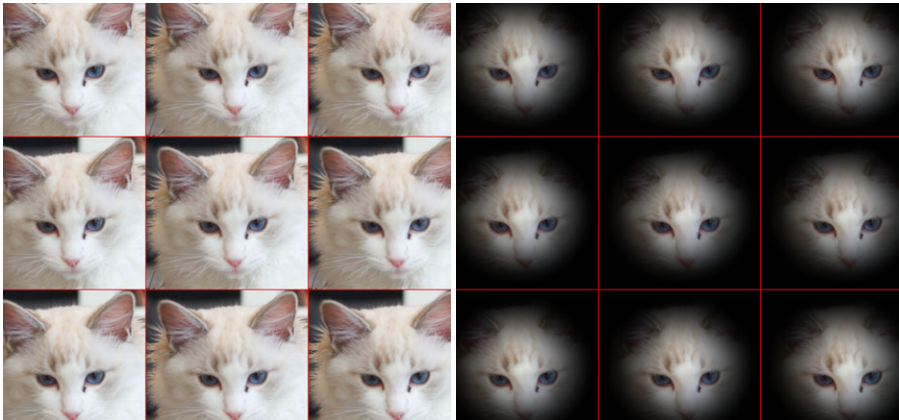


Figure 2.1: The circular correlation introduces boundary effects by repeating the image patch in a tiled pattern. Edges of the patch are marked in red. (left). These boundary effects can be mitigated by applying a windowing function to the image (left), avoiding discontinuities at the patch boundary.

For numerical reasons it is often useful to add a small regularization factor with respect to the coefficients of h , resulting in

$$\min_{h_t} \sum_{r,c} (x_t(r,c) \star h_t(r,c) - g(r,c))^2 + \lambda |h|^2, \quad (2.3)$$

where λ is the regularization. As the correlation operation results in a very large number of equations, solving the problem in this form is not tractable in general. If we assume that the correlation operation uses circular correlation the resulting equation systems has a circulant structure. The circular correlation operation is different only in that regions outside the image patch are assumed to be repetitions of the image patch itself, as shown in figure 2.1.

Circulant matrices are diagonalized by the Fourier transform [25]. This can be used to significantly reduce the computations required to find the template h . Using \hat{x} to denote the Fourier transform of x we can write the Fourier transform of (2.3) as

$$\min_h \sum_{u,v} (\hat{x}(u,v) \circ \hat{h}^*(u,v) - \hat{g}(u,v))^2 + \lambda |\hat{h}|^2. \quad (2.4)$$

Where $a \circ b$ is used to denote the element-wise¹ product between a and b , and x^* is the complex conjugate of x . The solution to this optimization can be found in the usual way of setting the derivative to zero followed by some algebraic manipulations, resulting in

$$\hat{h}(u,v) = \frac{\hat{g}^*(u,v) \circ \hat{x}(u,v)}{\hat{x}^*(u,v) \circ \hat{x}(u,v) + \lambda}. \quad (2.5)$$

¹This operation is also known as the Hadamard product

In order to reduce the influence of the boundary effects introduced by the circular convolution, a Hann window needs to be applied to the input image. As with the normalization this can be considered a pre-processing step. The rightmost image in figure 2.1 show the repeating pattern with this window applied.

Model updates

In practice, using only a single patch to create h results in an inferior appearance model, in most cases tracking will fail after only a few frames as the object appearance changes over time. For this reason it is important to include additional samples in the model. In most cases the patch used to initialize the tracker is the only available training data when tracking begins. Instead additional samples can be collected on-line by including the patches estimated by the tracker in each frame as additional data for the appearance model.

A simple approach for model updates is to use a rolling average scheme, as done in recursive filters [19]. In order to correctly handle the regularization term it is convenient to update the numerator and denominator of (2.5) separately. Using a forgetting factor of γ (or learning rate of $(1 - \gamma)$) the rolling average update for (2.5) is

$$A_{t+1} = \gamma \hat{g}^* \circ \hat{x}_t + (1 - \gamma)A_t \quad (2.6)$$

and,

$$B_{t+1} = \gamma \hat{x}_t \circ \hat{x}_t + (1 - \gamma)B_t, \quad (2.7)$$

where $\hat{h}_t = \frac{A_t}{B_t + \lambda}$. Applying this formulation to the detection in a new frame gives

$$\hat{y} = \frac{A_t \circ \hat{x}_t}{B_t + \lambda}, \quad (2.8)$$

where the score-map can be obtained by inverse Fourier transform of \hat{y} .

This concludes the description of the MOSSE tracker. The MOSSE is used as a base for most of the tracking work in this thesis. The papers A, B, E are direct extensions of this tracker, dealing with scale-estimation, integration of learned high-dimensional features and improved model update schemes respectively.

2.2 Extending the MOSSE tracker

Now that we have a baseline approach for tracking, it can be extended to handle various challenging situations in tracking. Unless the tracker is operating in an extremely controlled environment, such situations are common and difficult to avoid. Difficulties can be due to circumstances that make even

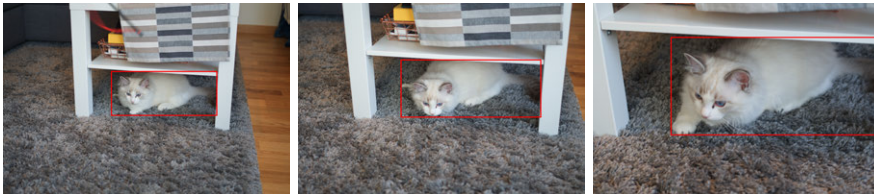


Figure 2.2: Cosmo the cat illustrating consequences of changing distance to the camera. As the camera moves closer, the percentage of the image occupied by Cosmo increases accordingly.

a perfect appearance model fit poorly to a particular image, such as motion blur or the tracked target undergoing dramatic changes in shape.

The perhaps simplest, and in many cases most successful modification that can be made to a tracker is to utilize more powerful feature representations. The MOSSE tracker can be extended to handle higher-dimensional features by a minor modification to the optimization problem (2.3), resulting in

$$\min_h \sum_{r,c}^{R,C} \left(\sum_{l=1}^L (h^l \star x^l)(r,c) - g(r,c) \right)^2 + \lambda \sum_l |h^l|^2, \quad (2.9)$$

where x^l is feature dimension l extracted from the image patch x . This extension requires corresponding modifications to the update rule, described in paper A.

Difficulties result from implicit or explicit assumptions of the tracker no longer holding. This can happen when an object moves out-of-frame, or some other object occludes it, leading to the object no longer being visible in the image. Other assumptions can be more subtle, such as the MOSSE filters fixed-size template resulting in an assumption of the target having constant size and shape in the image sequence.

Scale variations

One common situation that requires explicit handling by a tracker is that of apparent changes in object size. Such changes occur most commonly when the distance between the tracked object and the camera changes, as demonstrated in figure 2.2.

This is a problem for appearance models that are not inherently scale-invariant, such as the one used by the MOSSE filter. When the tracked object changes size drastically the updates to the appearance model of the MOSSE will cause the tracker to follow only a part of the target object, getting stuck on the background or fail altogether.

Successfully tracking in situations where scale-variations are present thus requires some explicit handling by the tracker algorithm itself. From a signal-processing perspective, an intuitive approach is to consider finding the correct

scale the same way as the position is found. That is, the MOSSE filter can be extended from the two dimensional translation case to a three dimensional translation-scale filter.

This can be done with a minor alteration to the MOSSE framework, by adding an additional scale-dimension to create a three dimensional filter. That is the optimization problem in (2.3) is modified to include an additional scale-dimension. The resulting filter operates in a three dimensional scale-translation volume. An illustration of this volume can be viewed as figure 2 in paper A. The scale-space volume is constructed by extraction of image patches with varying scale centered on the target, followed by interpolation to a desired size. Thus moving in the scale-direction of this volume is similar to zooming in or out on the target.

While this approach is elegant from a mathematical perspective, it is not very computationally efficient. This is due to the approach requiring three-dimensional filters over the cost-volume. Additionally it is required to repeat the feature extraction for each added scale. If advanced feature representations are used, this can be a significant computational burden as this step is often one of the most expensive parts of a tracking algorithm.

It is possible to avoid the explicitly construction of the scale-translation volume by evaluating the translation filter on several candidate patches of difference size [39]. That is (2.8) can be evaluated at a few of the scales that would otherwise be used to construct the scale-space volume. This avoids having to compute Fourier transforms in three dimensions, but the often more expensive feature extraction still needs to be repeated for each scale. The method is simple to implement however, and is often used in recent trackers such as the one described in paper E.

An approach that avoids both higher dimensional Fourier transforms, and repeated application of feature extractions is to introduce a separate scale-filter. This is similar to using separable filters for the joint scale-translation approach, with the difference that respective filters do not need to use the same feature representation. It is also possible to estimate the scale change after the translation has already been determined.

That is, we can first estimate the targets position in the image, then estimate the new scale at the updated position. Further, it is not required to utilize the same feature-representations for both filters. This allows the translation to be estimated using powerful high-dimensional features that might be computationally expensive to extract, while the scale estimation can be done using cheaper representations.

As shown in paper A, the scale filter works well when using gray-scale pixels as features. This saves a significant computational time compared to the other methods, as the feature extraction step only needs to be done once per frame. In practice the scale filter is learned using a one-dimensional variant of (2.9), where the pixels for each patch correspond to the feature-dimension.

Creating more robust appearance models

As noted in section 2.2 one of the most important components of a tracking algorithm is the feature representation used. Such representations can be either hand-crafted or learned. Of the hand-crafted features, the histogram of oriented gradients [7] (HOG) and the color names [54] [15] have been the most successful in the tracking setting.

The color names representation define a mapping from RGB color values, to a probability that the particular value would be described with one of eleven names for colors in the English language. The HOG feature representation are instead based on utilizing local normalizations of image-gradient features for different pixel regions. The features are constructed from such blocks into local orientation histograms.

Hand-crafted features typically have very efficient implementations, while producing compact descriptors of image data. Integrating such features into the DCF framework using the high-dimensional MOSSE filter described in (2.9) is straightforward for per-pixel features such as color names.

When using features whose grid-size is larger than single pixels some modifications to the detection step of the tracker is required. This can be done by scaling the obtained coordinate for the response peak using the grid-size of the features. However, this reduces the maximum resolution of the translation estimate to the resolution of the feature-grid. A more accurate approach that allows for more accurate translation estimates is to zero-pad the response y to the same size as original grid. By doing this before performing the inverse Fourier transform, the resulting response has the same resolution as the original pixel-grid. In practice this corresponds to interpolating the response using a trigonometric basis function. The same approach can be used to interpolate over scales in the scale-estimation step, where it is now possible to extract only a subset of the scales explicitly.

These hand-crafted features remained in use for significantly longer periods of time in the field of tracking than in other areas of computer vision. This is likely due to the requirements that tracking algorithms needs to remain relatively fast. This is a requirement both for real-time operation but also a consequence of the relative difficulty of training large scale models on tracking data.

While most tracking datasets contain large numbers of individual images, these images are highly correlated as they come from a relatively small set of sequences. As a comparison, object detection datasets contains thousands of images that are often selected to be as uncorrelated as possible. This situation makes it difficult to train image features directly on tracking data. Instead utilizing features extracted from networks learned on different tasks can be used. Paper B investigates the impact of utilizing features obtained from different layers of the VGG network [52].

The perhaps most surprising finding in paper B is that using feature representations from the first layers of the network is superior to tracking. This is in stark contrast to other tasks, where features from deeper layers of the network often provide better performance. A second surprise is that the increase in performance over using a combination of color names and HOG features in an otherwise identical tracker is relatively modest.

This is a consequence of the deeper feature layers having significantly lower resolution than earlier ones. While it is possible to up-sample the feature grid as described earlier, there are limits to how far the resolution of a coarse feature grid can be increased. As deeper layers of a neural network often reduce the resolution by half or more for each additional layer, the position estimation becomes too coarse to be useful in tracking.

While HOG-features are often used with a sub-sampling factor of 4 in tracking, neural network features are often reduced by a factor of two or more for each layer. In combination with the VGG network not being trained explicitly for tracking, but rather image classification [52], with object classes that only partially overlap with those in tracking lead to deeper layers having lower performance than the earlier.

Later work such as the CCOT [14] have since solved many of these issues in a tracking perspective, with further extension including fine-tuning of the representation on tracking data [49]. While these more advanced trackers outperform the methods in this thesis, they are to a large degree based on the same principles, either directly or through successive evolutions of a similar theoretical framework.

EVALUATION OF VISUAL OBJECT TRACKERS

In this chapter evaluation of visual tracking methods is discussed. The setting remains the same as in chapter 2, that is the evaluations are concerned with visual-object tracking, of a single target, in a causal setting. The evaluations are performed on image sequences of pre-recorded data, with hand-made annotations for the tracked object. The evaluation scores are derived from a comparison of the tracker output bounding boxes with the ground-truth for each frame.

Early tracking methods such as MOSSE [3] were evaluated only on relatively small, hand-picked datasets, or on subsets of larger datasets [8, 15]. Evaluations of this type are sufficient to reject completely unsound trackers, but in general do not provide an accurate assessment of a trackers performance in general. In order to more fairly compare methods, several different dataset initiatives have been proposed, such as the on-line tracking benchmark [56, 55], and the visual object tracking (VOT) challenge [31].

Of these two initiatives, the online-tracking (OTB) benchmark contains a larger number of videos. These videos are of highly varying quality, being a mix of sequences with high and low resolution, as well as color and grayscale data. A further issue is that a fair number of the sequences are comparatively easy, making them somewhat redundant. The target annotations provided are limited to axis-aligned bounding-boxes, where axis-aligned refers to the sides of the box being parallel with the image coordinate axes.

3.1 The visual object tracking challenge

The visual object tracking challenge is a yearly occurring evaluation of visual object trackers. The main challenge is concerned with short-term tracking using RGB data, with sub-challenges for tracking in thermal infrared, or

combinations of RGB and thermal infrared. The author of this thesis was a member of the technical committee for the years 2015 to 2017, where the 2017 challenge results is included as paper C.

The VOT challenge datasets [31, 38, 36, 35, 32, 34, 37, 33] take in many ways the opposite approach compared to the OTB benchmark. The VOT benchmark datasets contain a lower number of sequences, with each sequence being of higher difficulty. Even in the early VOT datasets all sequences used color-images, with relatively high-resolution. In order to increase the quality of the evaluation the dataset is updated each year, with new data or more fine grained annotations. The early years [31, 38] used axis-aligned bounding box representations. These were later replaced by general bounding boxes [36, 35, 32, 34, 37], and most recently per-pixel segmentation masks [33].

The VOT benchmark evaluates trackers in terms of two main criteria: accuracy and robustness. The accuracy is calculated in terms of the intersection over union (IoU) between the prediction produced by the tracker and the ground-truth. The robustness measures resistance to failures, where a tracker is considered to have failed if the IoU with the ground-truth is zero.

The IoU is a measure of the similarity of two sets. The similarity score is between 0 (the set intersection is empty) to 1 (the two sets are identical). It was originally proposed by Jaccard [28] as a way to compare the biomes of different alp-tops. For this reason it is sometimes known as the Jaccard index. It was introduced as a way to compare the bounding boxes produced by object detectors with the respective ground-truth in the pascal challenge [17], where a detection is considered successful if the IoU exceeds 0.5.

For two axis-aligned bounding boxes A and B the intersection-over-union (IoU) score is defined as

$$\text{IoU} = \frac{\text{area}(A \cap B)}{\text{area}(A \cup B)}, \quad (3.1)$$

where the \cup and \cap operators denote union and intersection regions of the two boxes respectively. Note that using this definition the area of the union region is most easily calculated as

$$\text{area}(A \cup B) = \text{area}(A) + \text{area}(B) - \text{area}(A \cap B), \quad (3.2)$$

as the region $A \cup B$ is not itself an axis aligned bounding box. A visual example of this can be found in 3.1.

In the VOT evaluation a tracker is initialized on the first frame using the ground-truth bounding-box. The tracker then tracks the object in each frame of the sequence until a failure is detected. In practice many trackers, such as those described in chapter 2 are restricted to tracking axis-aligned bounding boxes. Such methods are required to adapt the annotation used for initialization by the VOT toolkit themselves.

If a tracker has zero IoU with the ground-truth in a frame, it is considered to have failed. The toolkit will then restart the tracker a few frames

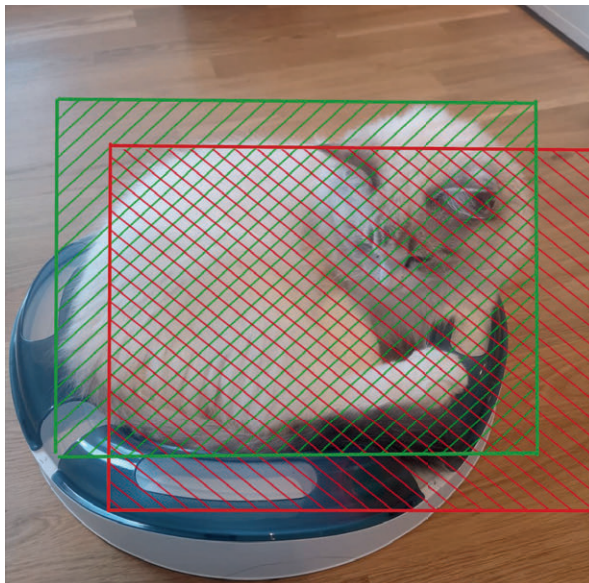


Figure 3.1: Cosmo the cat and two bounding boxes. The green marked region corresponds to a typical ground-truth bounding-box, and the red to a somewhat failed detection. The intersection region is covered by both boxes, and the union is the total area covered by any box.

after the failure was detected. This is done in order to make maximal use of limited annotated data, as well as in order to allow more than a single failure per sequence. This avoids the earliest frames in a sequence having a disproportionate influence on the final overlap. This gives the robustness as the rate-of-failure for a tracker, where a robustness of 1 corresponds to never failing and, a robustness of 0 corresponds to failing every frame. The accuracy is the mean intersection over union for each of the tracked frames.

In practice accuracy and robustness is often related, in that a more accurate tracker is likely to be more robust as well. This is itself likely a consequence of the fact that since most trackers utilize online-updated appearance models for the target, model drift is a major concern. With more accurate tracking, model drift is less likely to occur as the additional collected data is of higher quality. The accuracy versus robustness scores can be plotted on a 2D plane, as demonstrated for some trackers from the 2017 VOT challenge in figure 3.2.

The ants1 sequence has most of the methods performing fairly well, with several having zero failures throughout the sequence. As seen in figure 3.3, the sequence recorded using a fixed camera above a petri dish containing several ants. These ants are virtually identical except for a small marking. As the view is a top-down one with objects moving in a plane, trackers do not need

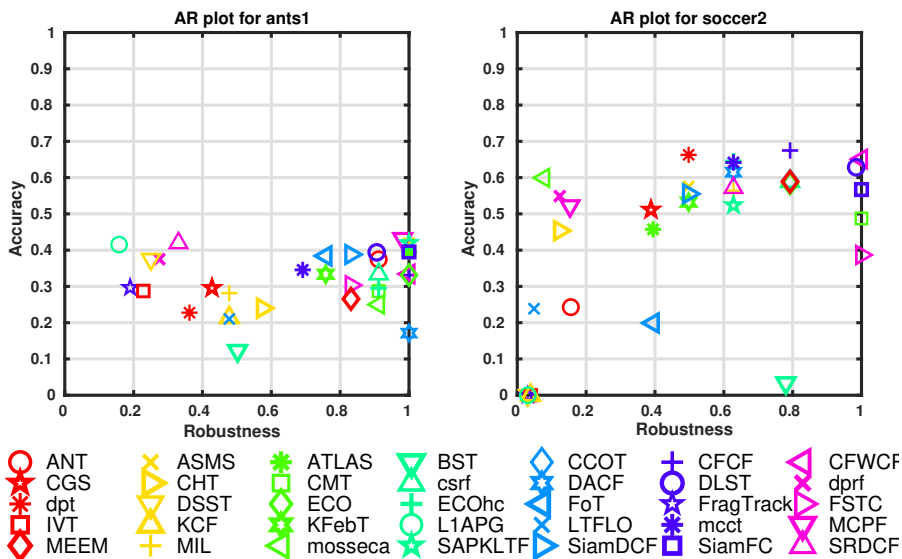


Figure 3.2: Accuracy-Robustness plot for a few of the trackers from the 2017 VOT challenge on the ants1 and soccer2 sequences. The x-axis corresponds to better accuracy and the y-axis to better robustness. The general tracking performance is highest in the top-right corner.

to be scale-adaptive. It is however important that they are robust to in-plane rotations, as ants switching direction will result in a rotation from the point of view of the camera.

In the soccer2 video, failures are instead common, with several trackers failing on most frames. While this sequence lacks in-plane rotations, it contains significant motion blur and fast motions. This results in many trackers either failing instantly, or having low accuracy scores. For trackers that are resistant to this particular mode of failure, the sequence is easier and higher accuracy and robustness is achieved.

3.2 Issues with using intersection over union scores

Unfortunately, using a criterion of IoU being 0 to detect tracker failures is problematic. When compared to the object detection setting, a detection is often considered successful that has more than 0.5 IoU with the ground-truth bounding box. In most cases 0.5 overlap can be considered a fairly poor either in terms of localization or size of the tracked target.

A particularly serious issue was discovered when evaluating trackers during the 2015 VOT-TIR sub-challenge of VOT. One of the submitted trackers had comparatively high robustness, but unreasonably low accuracy. Upon



Figure 3.3: A few frames from the ants1 and the soccer2 sequences in the VOT 2017 dataset. Black bounding box is the ground-truth, red is from the DSST tracker. In the ants figure tracking fails due to the in-plane-rotation, while in the soccer case the sudden movement of the camera results in low accuracy.

inspection it was noticed that the tracker in question had a tendency to output very large bounding-boxes, sometimes covering the entire image.

This meant the tracker largely avoided failures, as the overlap with the ground-truth bounding never reached zero even when tracking was in a wider sense unsuccessful. While one might expect such behavior to severely penalize the accuracy, there are circumstances where the IoU metric does not behave as expected.

The reason for this can be best described by considering the IoU score in terms of image regions being correctly classified or not. With A as the ground-truth, the region $A \cap B$ is then the region of true positive (TP) predictions. The false negative (FN) is the part of A that is not in $A \cap B$ and the false positive (FP) the corresponding parts of B .

Using these terms (3.1) is

$$\text{IoU} = \frac{\text{area}(TP)}{\text{area}(TP + FP + FN)}. \quad (3.3)$$

Conspicuously missing from this expression is what often is the largest part of the image. That is the part not covered by any bounding box. That is the region correctly classified as not belonging to the target, the true negative (TN). In most cases this is the majority of the image.

The consequence is that the penalty for over-estimating the bounding box is less severe than when under-estimating it. This is a result of the fact that over-estimation, as long as the position is approximately correct often result in the TP region increasing in size. Increasing the TP term will come with a corresponding decrease in the FN term. If the bounding box is instead expanded in a way that only includes additional background, the only term that changes is the FP term.

Taking the extreme case of a tracker always outputting a bounding box that covers the majority of the image the IoU metric will never be zero. In fact the main influences on the IoU will be what percentage of the image is covered by the tracked object. This is due to the TP term having its maximum value, while the FN term is zero.

If one considers only infinite-sized images, this is not a problem, as the IoU will approach zero if the bounding box approaches infinite size. However, in practice images have finite-size, meaning that there is an upper bound for how large it is meaningful to make the prediction. This can be taken advantage of by outputting a bounding box with massively larger than desired size.

Paper D proposes a way to avoid over-estimating the performance in such scenarios. Based on modify the IoU score in order to include the TN term in the calculation. This can be done by considering the inverse IoU, that is the IoU with respect to the tracker output, and the complement to the ground-truth. By adding these two terms together with a weighting factor, a new IoU score can be created as

$$\text{IoU}_2 = \omega_{bg} \frac{\text{area}(TP)}{\text{area}(TP + FP + FN)} + \omega_o \frac{\text{area}(TN)}{\text{area}(TN + FN + FP)}. \quad (3.4)$$

Where ω_o and ω_b is the relative area covered by the object annotation, and the background respectively. As shown in paper D this approach for overlap-estimation successfully penalizes naive trackers, even in cases where the tracked object covers the majority of the image.

In practice situations where the tracked object covers the majority of the image are rare, making this modification to the IoU score closer to a solution in search of a problem, than a critical fix to a common issue. This is even more true as the vision community moves away from coarse bounding-box representations to more fine-grained ones such as per-pixel segmentations or bounding-polygons.

TRACKING IN PRACTICE AND UNCERTAINTY

This chapter deals with common issues that occur when tracking in real-world scenarios. Some major difference to the evaluations described earlier is that the tracking system is often required to initialize itself, along with needing to tracker over longer sequences than is common in benchmark datasets.

As ground-truth annotations are often not available for initialization or re-setting of a failed tracker, these tasks need to be accomplished by the tracker itself. An additional problem is that when no ground-truth is available, detecting that a tracker has failed is a difficult problem in itself. While most trackers output some form of per-pixel score, it is not easy to interpret this score as a measure of tracking performance.

Instead, other approaches must be used. One of these is to utilize an external system for monitoring the behavior of the appearance model used by the tracker. This approach is taken in paper F where the output of the tracker is compared with a pre-trained object-detector and a Kalman filter. While this approach is successful for recovering from tracking failure, the system is restricted to work on object types for which training data is available for creating the detector.

An alternative approach based on continuously re-weighting the samples used for the trackers appearance model is proposed in paper E. In this paper the SRDCF [13] tracker is extended in order to avoid model drift by retroactively re-weighting collected samples in each frame. This allows for retroactively deciding that a collected sample was collected due to a low-quality prediction.

Ideally, we would be able to decide right away if a prediction is of low quality. This requires estimating a predictions uncertainty along with the prediction itself. An approach for this is investigated in Paper G. Here the complexity of the tracking problem is reduced by tracking point-targets in

a setting with known geometry. Paper G investigates obtaining uncertainty measures along with predictions of stereo disparity by predicting a distribution rather than point-estimates.

4.1 Tracking in practice

This section describes tracking in real world settings, where no human initialization is provided, and relatively long time-periods including long-term occlusions are included. In order to increase the level of autonomy a system is capable of it is therefore important to be able to initialize the tracking of interesting objects automatically.

However, as many object-detectors are quite processing intensive, it is likely not possible to detect objects in every image, particularly if processing is required to be done on-board. While it is possible to stream video data to an external computer, this often requires either a very high bandwidth link between the robot and the external computer or utilization of heavily compressed video.

Even if the communication line between the robot and the computing platform has sufficient bandwidth to stream the image data, it will introduce a certain amount of latency. This can be a problem if the output of the vision algorithm is used as input to a control system. While on-board processing is often more limited in hardware, as it is often limited by being battery powered, it offers significantly lower latencies and is in general more reliable.

Long term tracking of humans

In long-term tracking scenarios, visual trackers using online-learning, such as those presented in 2 often suffer from issues of model drift. As the tracker model is updated with data gathered while tracking, unexpected situations can cause these updates to corrupt the model. Most often this results in either erratic behavior or the tracker eventually switching to follow something other than the intended target. While this issue occurs in short-term settings as well, it is less of an issue as the sequence often ends before the tracking fails.

Model drift can be caused by a number of situations such as occlusions causing the tracker to get stuck on the occluding object, faulty or absent scale-estimation, presence of distractor objects or minor errors in tracking compounding over time. Of these issues, scale-estimation has already been discussed in chapter 2 as well as paper A.

Dealing with occlusions can be done in a number of ways. The simplest being to pretend that the tracked object is never occluded. This works in many cases, as occlusions are often short-term or only covers part of the object. Situations involving longer term occlusions are more difficult, as the tracked object often goes out of view in one part of the image, only to become visible in a different part.

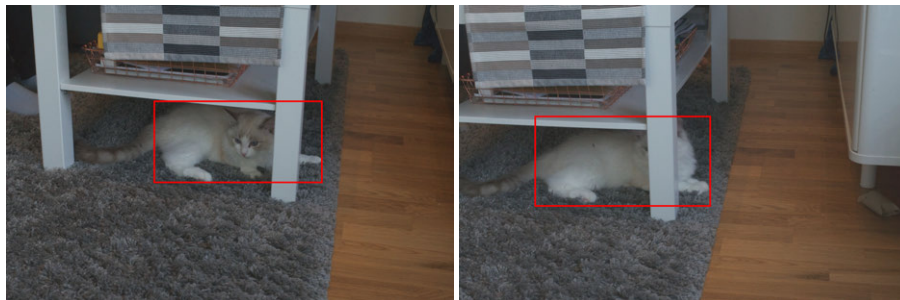


Figure 4.1: Cosmo the cat demonstrates the difficulties with occlusion. While Cosmo is not fully occluded, there is a significant risk of the appearance model adapting to track the table instead of the cat, particularly in situations such as the left image

In such situations it is often difficult to even know that the tracked object is no longer visible. One way of detecting such occurrences that is used in paper F is to monitor the peak-to-sidelobe ratio [3] of the trackers response, using this as an estimate of detections certainty.

If the tracked objects type is known, it is possible to utilize a trained object detector to verify that there is still an object of the expected type in the region output by the tracker. This will work as a way to detect occlusions as long as the occluding object is not of the same type as the tracked one. Paper F utilizes a combination of these two approaches together with a Kalman-filter for increasing the robustness of a tracker over longer term tracking.

Detecting objects to track

Object detection is a problem closely related to tracking, with the goal is to detect all objects of a given class. While trackers presented in this thesis are in some sense object detectors, they are trained to detect only a specific object instance, and generalization beyond this one object is not desirable.

General object detectors are typically trained off-line using large datasets of annotated images. As they are required to detect an unknown number of objects in each image, the detection scores are often thresholded before outputting any predictions. Most recent object detectors [48, 47, 16] are based on deep learning approaches. As the hardware used for the system described in paper F was limited and did not have GPU, these methods did not suit the hardware available at the time.

For these reasons, the system described in paper F utilize a simpler detector based on HOG features and a linear support vector machine [7]. This approach requires significantly less computational power than the deep-learning based methods, but will in practice produce detection of lower quality.

Adding an object detector into a tracking framework also allows the system to recover in case the tracking fails completely. By periodically storing the appearance model used by the tracker, it can be used to verify if a newly detected object is the same as one seen before by evaluating the detection step of the tracker on each new detection. This approach allows resuming tracking of a target in case it is occluded for long periods or temporarily moves out of frame.

While this approach works fine in practice, it is limited to tracking objects of a known type. Another downside with the system presented in F is that it requires complex logic and a large number of hand-tuned parameters in order to make it behave correctly. A more convenient approach would be to include the reliability of each sample in the optimization procedure.

4.2 Retroactively weighting samples

In the formulation of the DCF model update from chapter 2 each sample has a weight that is exponentially decaying as additional samples are added. This is a consequence of the linear interpolation resulting in exponential down-weighting of older samples as additional data is added.

The model at time t is constructed from samples whose weight α_k is exponentially decaying as additional samples are added. Using this, the problem as in 2.3 can be reformulated to

$$\min_{h_t} \sum_k^t \alpha_k \sum_{r,c} (x_t(r,c) \star h_t(r,c) - g(r,c))^2 + \lambda|h|^2, \quad (4.1)$$

where the sample weights are fixed.

In the original formulation the weight for each sample is not included in the optimization process, and the α_k for each sample is set by the forgetting factor. This is a consequence of the recursive update. While this is practical in that it requires storing only the current model and allows for a memory of theoretically infinite length, it has a major disadvantage in that it always gives the oldest samples the lowest weight.

When the tracker is initialized by a human annotation, the result is that the only known good sample has the lowest weight. An additional disadvantage is if the tracked object is occluded, as the samples depicting the occluding object will be given higher weight than the object itself.

This can be alleviated by dynamically setting the weight of each sample in every frame. By reformulating the optimization problem in (4.1) to include the sample weights, it is possible to retroactively down-weight samples containing occlusions or significant motion blur. Re-optimizing the weights in each frame also allows for the weights to change as additional data is collected.

Continually re-weighting the collected samples requires solving the following optimization problem in each frame. By defining $L(h, x, g) = \sum_{r,c} (x_t(r,c) \star$

$h_t(r, c) - g(r, c))^2$ an optimization problem that includes the sample weights can be defined as

$$\min_{h, \alpha} \sum_{k=1}^t \alpha_k L(h, x_k, g) + \frac{1}{\mu} \sum_{k=1}^t \frac{\alpha_k^2}{\beta_k} + \lambda |h|^2 \quad (4.2a)$$

$$\text{subject to: } \alpha_k \geq 0 \quad (4.2b)$$

$$\sum_k^t \alpha_k = 1 \quad (4.2c)$$

Where the β_k correspond to a prior weight for each sample, and μ is a parameter for giving more or less weight to the prior versus the data itself. This prior can be set to the same exponential forgetting curve as in the original formulation, as it is in most cases correct that the most recent sample is the most similar to the next detection. Paper E proposes a tracker based on the SRDCF [13] using this approach.

While this approach allows the tracker to retroactively weight detections, it will in most cases take a few frames to notice if a sample was particularly bad. An example of this is shown in figure 2 of paper E. Ideally we would know as a detection happens if it is of high or low quality, as this means we can use or discard it directly.

4.3 Recognizing image regions with uncertainty

In the previous parts of this chapter the ways to deal with detections that are of low quality can only be done retroactively when additional data is gathered. This means that when a detection is made, we have still have no good estimate of how reliable the detection is. One attempt at addressing this is proposed in Paper G in a stereo disparity setting. Estimation of stereo disparity can be viewed as a simplified form of tracking, using constraints derived from projective geometry to limit the possible positions for the sought after region.

The visual tracking task can be simplified by considering only a pair of images, that is we can avoid creating appearance models that are updated over time. The task can be further simplified by considering only targets that are pixel-sized, this avoids having to estimate changes in size or shape. If we track all pixels in the image individually for a pair of images the result is the optical flow between the two images [53].

Estimating the optical flow between two images requires that the translation for each pixel is determined with respect to movement in both row and column dimensions of the image. If the geometry between the two cameras, is known, this can be reduced to a single degree of freedom [53, 24]. With known geometry between the two cameras the image of a 3D point in the first image is guaranteed to lie along the points epipolar line in the second image. A visualization of this can be seen in figure 4.2. The red circle in the left image corresponds to the red line in the right image.

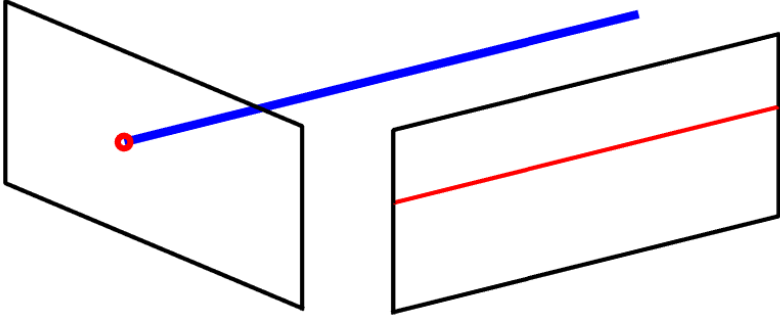


Figure 4.2: Visualization of epipolar geometry. Any point along the blue line will be projected to the red circle in the left image. The blue line is projected into the right image as the red line. This means that in order to track the red point in the right image, it is sufficient to search along the red line.

This can be reduced to a single degree of freedom by taking advantage of well known geometrical constraints between projections of a single point into two cameras [24] where the relative configuration of the cameras is known.

Stereo disparity estimation

Using images from two synchronized cameras, with known relative calibration it is possible to estimate the distance for each point in one image using the disparity of the points. The disparity d can be converted to a distance by using the focal length f and the baseline b between the cameras

$$D = \frac{fb}{d}, \quad (4.3)$$

where D is the distance from the camera to the 3D point [53, 24]. This information is useful in many situations requiring knowledge of the three-dimensional structure of a scene such as mapping, localization or navigation. In this thesis, the cameras are assumed to be placed horizontally, and image points are tracked from the left camera to the right.

If the images used are rectified then the epipolar lines are also parallel. This means that for each point (r, c) in the leftmost image the corresponding point can be found at a position $(r, c-d)$ where d is the disparity of the point. Due to this regularity, it is possible to effectively enumerate each possible disparity by stacking shifted copies of the left and right images in a three-dimensional volume [5, 30]. A visualization of such a volume can be viewed as

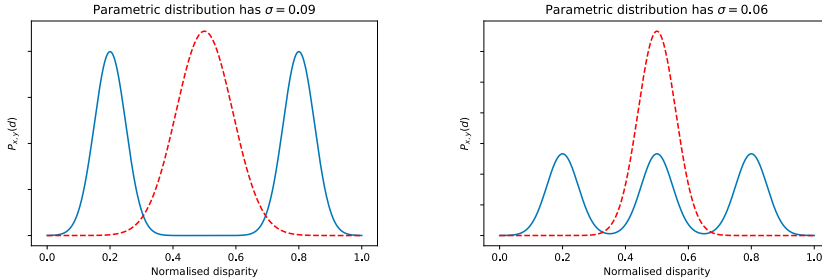


Figure 4.3: An attempt to fit a parametric Gaussian to multi-modal data. In the left figure two possible disparities exist, making the mode of the distribution useless. In the right figure the three modes cause the variance of the prediction to decrease, while in practice situations like this might not have well defined disparity.

figure 1 of paper G. The matching volume can be indexed by the coordinates (r, c, d) , where a step in the d direction corresponds to moving one step to the left in the image we search for the match.

The assumption that the projection of a corresponding point can be found along the epipolar line will not hold if the point is not visible in both images. From the perspective of tracking image regions, there does not exist a reasonable disparity for such pixels. While it is possible to determine what the disparity should be using (4.3), this requires knowledge of the distance to be obtained from some other source. For this reason, knowing the uncertainty for each prediction, becomes useful since a system using the estimated distances can ignore those measurements that have high uncertainty.

Approaches for estimating this uncertainty include left-right consistency for the estimates, analysis of local or global properties of the scores assigned to each disparity [26, 44] as well as modifications to methods to output distributions rather than point-estimates [27, 18], where a parameter related to the width of a distribution is used as a measure of uncertainty.

Most approaches based on predicting distributions are limited to predicting the parameters for distributions of known shape [27, 18]. In practice this can restrict the shape of the predicted distributions to a form that does not accurately reflect the type of errors that occur in recognition problems. This can be an issue since it is common for images to have multiple objects of similar appearance, such as the ants in figure 3.3, or repeated visual patterns such as those from a fence or tree-trunks.

In situations of this type the variance of a unimodal distribution can even be shown to decrease if additional modes are added to the distribution. One such example can be seen in figure 4.3, where the mean of the distribution

is either a very low quality prediction, or the variance fails to represent how ambiguous the data is.

This problem can be avoided by using a representation for the predicted probability density that is not limited to a specific shape, such as a density mixture or kernel density estimator. However such representations still require choosing the number of components and shape of the basis functions. This can be avoided by instead learning the shape of the representation as a neural network. As neural networks impose few restrictions on the functions they represent, this avoids the problems of using a representation of fixed shape.

Many current methods for disparity estimation are based on neural networks, but are trained to output only point-predictions [5, 30, 58, 57], as this is shown to provide a better disparity estimation performance than when a cross-entropy loss is used during training [30]. This is likely due to the direct regression approach minimizing the L^1 error between the predicted and ground-truth disparities. The L^1 error being distance between the predicted disparity and the ground-truth.

If the cross-entropy between the two distributions is minimized instead of the L^1 error then this geometric interpretation is lost. This can be easily shown by considering the cross-entropy for a prediction q and a ground-truth distribution p

$$H = - \sum_{i=1}^N p_i \log q_i. \quad (4.4)$$

If both p and q are point-masses, $p = \delta_k$, and $q = \delta_n$, the cross entropy is minimized for $k = n$. It is maximal for any $k \neq n$, that is the distance between n and k does not matter. While this is suitable for classification problems where the distance between all classes can be considered equal, it is less suitable when predicting a position or offset as is done for disparity.

One measure between the similarity of distributions that includes a geometric interpretation is the Wasserstein or earth-movers distance

$$W^1(p, q) = \inf_{\gamma \in \Gamma(p, q)} \int_{-\infty}^{\infty} |z - w| d\gamma(z, w), \quad (4.5)$$

where Γ is all joint distributions with marginal distribution p and q . The earth-movers distance measures the similarity between two distributions by considering the amount and distance of probability mass that needs to be moved to turn q into p . Using this loss function predictions that are far away from the ground-truth are penalized more severely.

In general the Wasserstein distance requires the solution of an optimal transport problem [6, 43]. In the case of a one dimensional distribution there exists a solution in terms of the cumulative distributions for p and q [46]. This effectively allows learning the parameters of a network that can output a general distribution, as well as using a general ground-truth distribution. Returning to the example of two point-masses, can be shown to be $W^1(\delta_k, \delta_n) = |k - n|$, preserving the geometry of the original problem.

What remains is then to decide on a measure for the uncertainty of the predicted distribution. While it is likely that the predictions will have a single mode in many cases, meaning variance is usable, it will fail in situations where the prediction is multi-modal. The variance also lacks a convenient way to represent the idea of there being no good match, or that all q are equally bad predictions requires the variance to be infinite. The Shannon entropy [51] avoids both of these problems

$$E = - \sum_i^N p_i \log p_i. \quad (4.6)$$

The entropy is zero when all probability is focused in a single p_i , as a consequence of p_i or $\log p_i$ being zero for any i . The entropy is maximized when $p_i = \frac{1}{N} \forall i$ [51]. For situations such as those in figure 4.3 the entropy will increase with additional peaks [51, 42, 2].

That is, by using E as a measure of uncertainty we can be certain that the uncertainty increases with a more ambiguous prediction, as well as encode the situation where no good disparity exists by using the uniform distribution as a label for such pixels.

Paper G to train a deep neural network to predict a distribution over disparities. The parameters of the network are found by minimizing the Wasserstein distance between the predicted distribution and a ground-truth distribution. Uncertainties are obtained from the predicted distribution as its entropy. For pixels where the disparity is not well defined the uniform distribution is used as ground truth, this allows the network to better predict regions where the prediction is likely to be wrong.

CONCLUDING REMARKS

This thesis has presented several methods for visual object tracking. These range from extensions to deal with variations in object size, to introducing learned feature representations into the tracking framework, to adaptively rejecting collected samples, as well as real-world implementations where tracking is part of a greater system.

While the Fourier domain approach for updating the appearance model in the trackers have been largely replaced by approaches taking advantage end-to-end learning of most stages of the tracking pipeline, many of the current methods share a great deal of the design with the tracking methods described in this thesis. A cursory look at the submissions for the most recent VOT challenge reveals that there is still a large number of correlation filter based trackers submitted.

In a wider sense, the general problem of tracking objects is still a very active research area. As the field of computer vision becomes increasingly dependent on using large-scale datasets for evaluations it is important to also continue to investigate and improve on the evaluation criteria and benchmark datasets in order to avoid creating methods that only solve the benchmark problem, or a accidentally over-fit to corner cases in the evaluation metric.

The difficulties that arise in robotics scenarios are in many ways resulting from lack of ability to combine the outputs from different algorithms into a coherent framework where components can correct one-another. One step in this direction is to utilize algorithms that can output some estimate of their own reliability. Estimating uncertainties along with predictions can be conveniently done by having methods produce distributions instead of point-estimates as predictions. There is however a great deal of work left to both in uncertainty estimation and more generally in building larger systems where individual components have parameters learned from data.

BIBLIOGRAPHY

- [1] M. S. Banks and P. Salapatek. “Acuity and contrast sensitivity in 1-, 2-, and 3-month-old human infants.” In: *Investigative Ophthalmology & Visual Science* 17.4 (1978), pp. 361–365.
- [2] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. “Visual object tracking using adaptive correlation filters.” In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 2544–2550.
- [4] E. A. Capaldi, A. D. Smith, J. L. Osborne, S. E. Fahrback, S. M. Farris, D. R. Reynolds, A. S. Edwards, A. Martin, G. E. Robinson, G. M. Poppy, et al. “Ontogeny of orientation flight in the honeybee revealed by harmonic radar.” In: *Nature* 403.6769 (2000), pp. 537–540.
- [5] J.-R. Chang and Y.-S. Chen. “Pyramid stereo matching network.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5410–5418.
- [6] M. Cuturi. “Sinkhorn distances: Lightspeed computation of optimal transport.” In: *Advances in neural information processing systems* 26 (2013), pp. 2292–2300.
- [7] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection.” In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.
- [8] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. “Accurate scale estimation for robust visual tracking.” In: *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press. 2014.
- [9] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. “Coloring channel representations for visual tracking.” In: *Scandinavian Conference on Image Analysis*. Springer. 2015, pp. 117–129.

- [10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. “Discriminative scale space tracking.” In: *IEEE transactions on pattern analysis and machine intelligence* 39.8 (2016), pp. 1561–1575.
- [11] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. “Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1430–1438.
- [12] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. “Convolutional features for correlation filter based visual tracking.” In: *Proceedings of the IEEE international conference on computer vision workshops*. 2015, pp. 58–66.
- [13] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. “Learning spatially regularized correlation filters for visual tracking.” In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4310–4318.
- [14] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. “Beyond correlation filters: Learning continuous convolution operators for visual tracking.” In: *European conference on computer vision*. Springer. 2016, pp. 472–488.
- [15] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer. “Adaptive color attributes for real-time visual tracking.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1090–1097.
- [16] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. “Centernet: Keypoint triplets for object detection.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6569–6578.
- [17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. “The pascal visual object classes (voc) challenge.” In: *International journal of computer vision* 88.2 (2010), pp. 303–338.
- [18] J. Gast and S. Roth. “Lightweight probabilistic deep networks.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3369–3378.
- [19] F. Gustafsson, L. Ljung, and M. Millnert. *Digital Signalbehandling*. Studentlitteratur, 2001.
- [20] G. Häger, G. Bhat, M. Danelljan, F. S. Khan, M. Felsberg, P. Rudl, and P. Doherty. “Combining visual tracking and person detection for long term tracking on a uav.” In: *International Symposium on Visual Computing*. Springer. 2016, pp. 557–568.

- [21] G. Häger, M. Felsberg, and F. S. Khan. “Countering bias in tracking evaluations.” In: *13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, January 27-29, Funchal, Madeira*. Vol. 5. Science and Technology Publications, Lda. 2018, pp. 581–587.
- [22] G. Häger, M. Persson, and M. Felsberg. “Predicting disparity distributions.” In: *2021 International Conference on Robotics and Automation (ICRA) (in print)*. 2021.
- [23] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr. “Struck: Structured output tracking with kernels.” In: *IEEE transactions on pattern analysis and machine intelligence* 38.10 (2015), pp. 2096–2109.
- [24] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, 2004. DOI: 10 . 1017 / CB09780511811685.
- [25] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [26] X. Hu and P. Mordohai. “A quantitative evaluation of confidence measures for stereo vision.” In: *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012), pp. 2121–2133.
- [27] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox. “Uncertainty estimates and multi-hypotheses networks for optical flow.” In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 652–667.
- [28] P. Jaccard. “The distribution of the flora in the alpine zone. 1.” In: *New phytologist* 11.2 (1912), pp. 37–50.
- [29] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems.” In: *Transactions of the ASME—Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.
- [30] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. “End-to-end learning of geometry and context for deep stereo regression.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 66–75.
- [31] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, T. Vojir, A. Gatt, A. Khajenezhad, A. Salahledin, A. Soltani-Farani, A. Zarezade, A. Petrosino, A. Milton, B. Bozorgtabar, B. Li, C. S. Chan, C. Heng, D. Ward, D. Kearney, D. Monkosso, H. C. Karaimer, H. R. Rabiee, J. Zhu, J. Gao, J. Xiao, J. Zhang, J. Xing, K. Huang, K. Lebeda, L. Cao, M. E. Maresca, M. K. Lim, M. El Helw, M. Felsberg, P. Remagnino, R. Bowden, R. Goecke, R. Stolkin, S. Y. Lim, S. Maher, S. Poullot, S. Wong, S. Satoh, W. Chen, W. Hu, X.

- Zhang, Y. Li, and Z. Niu. “The Visual Object Tracking VOT2013 Challenge Results.” In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 98–111. DOI: 10.1109/ICCVW.2013.20.
- [32] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin Zajc, T. Vojir, G. Häger, A. Lukežic, A. Eldesokey, et al. “The visual object tracking VOT 2017 challenge results.” In: *Proceedings of the IEEE international conference on computer vision workshops*. 2017, pp. 1949–1972.
- [33] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Č. Zajc, M. Danelljan, A. Lukežic, O. Drbohlav, L. He, Y. Zhang, S. Yan, J. Yang, G. Fernandez, and et al. *The Eighth Visual Object Tracking VOT2020 Challenge Results*. 2020.
- [34] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Č. Zajc, T. Vojir, G. Bhat, A. Lukežic, A. Eldesokey, G. Fernandez, and et al. *The sixth Visual Object Tracking VOT2018 challenge results*. 2018.
- [35] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Č. Zajc, T. Vojir, G. Häger, A. Lukežič, and G. Fernandez. *The Visual Object Tracking VOT2016 challenge results*. Springer. Oct. 2016. URL: <http://www.springer.com/gp/book/9783319488806>.
- [36] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernandez, T. Vojir, G. Häger, G. Nebehay, and R. Pflugfelder. “The visual object tracking vot2015 challenge results.” In: *Proceedings of the IEEE international conference on computer vision workshops*. 2015, pp. 1–23.
- [37] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Č. Zajc, O. Drbohlav, A. Lukežic, A. Berg, A. Eldesokey, J. Kapyla, and G. Fernandez. *The Seventh Visual Object Tracking VOT2019 Challenge Results*. 2019.
- [38] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Č. Zajc, G. Nebehay, T. Vojir, G. Fernandez, A. Lukežič, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangeršič, G. Häger, F. S. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Y. Choi, J.-W. Choi, J. F. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Öfjäll, K. M. Yi, L. Qin, L. Wen, M. E. Maresca, M. Danelljan, M. Felsberg, M.-M. Cheng, P. Torr, Q. Huang, R. Bowden, S. Hare, S. Y. Lim, S. Hong, S. Liao, S. Hadfield, S. Z. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, and Z. Niu. *The Visual Object Tracking VOT2014 challenge results*. 2014. URL: <http://www.votchallenge.net/vot2014/program.html>.
- [39] Y. Li and J. Zhu. “A scale adaptive kernel correlation filter tracker with feature integration.” In: *European conference on computer vision*. Springer. 2014, pp. 254–265.

-
- [40] B. D. Lucas, T. Kanade, et al. “An iterative image registration technique with an application to stereo vision.” In: Vancouver, British Columbia. 1981.
- [41] L. Matthews, T. Ishikawa, and S. Baker. “The template update problem.” In: *IEEE transactions on pattern analysis and machine intelligence* 26.6 (2004), pp. 810–815.
- [42] K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2021. URL: probml.ai.
- [43] G. Peyré, M. Cuturi, et al. “Computational optimal transport: With applications to data science.” In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.
- [44] M. Poggi, F. Tosi, and S. Mattoccia. “Quantitative evaluation of confidence measures in a machine learning world.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5228–5237.
- [45] D. A. Pomerleau. *Alvinn: An autonomous land vehicle in a neural network*. Tech. rep. 1989.
- [46] A. Ramdas, N. G. Trillos, and M. Cuturi. “On wasserstein two-sample testing and related families of nonparametric tests.” In: *Entropy* 19.2 (2017), p. 47.
- [47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You only look once: Unified, real-time object detection.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [48] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>.
- [49] A. Robinson, F. J. Lawin, M. Danelljan, F. S. Khan, and M. Felsberg. “Learning fast and robust target models for video object segmentation.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7406–7415.
- [50] C. A. Rosen and N. J. Nilsson. *Application Of Intelligent Automata to Reconnaissance*. Tech. rep. Stanford Research Institute, Oct. 1966.
- [51] C. E. Shannon. “A mathematical theory of communication.” In: *Bell system technical journal* 27.3 (1948), pp. 379–423.

- [52] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.
- [53] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [54] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus. “Learning color names for real-world applications.” In: *IEEE Transactions on Image Processing* 18.7 (2009), pp. 1512–1523.
- [55] Y. Wu, J. Lim, and M.-H. Yang. “Object Tracking Benchmark.” In: *PAMI* (2015).
- [56] Y. Wu, J. Lim, and M.-H. Yang. “Online Object Tracking: A Benchmark.” In: *CVPR*. 2013.
- [57] H. Xu and J. Zhang. “Aanet: Adaptive aggregation network for efficient stereo matching.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1959–1968.
- [58] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr. “Ga-net: Guided aggregation net for end-to-end stereo matching.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 185–194.
- [59] J. Zhang, S. Ma, and S. Sclaroff. “MEEM: robust tracking via multiple experts using entropy minimization.” In: *European conference on computer vision*. Springer. 2014, pp. 188–203.

PART II

PUBLICATIONS

Papers

The papers associated with this thesis have been removed for copyright reasons. For more details about these see:

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-175177>

FACULTY OF SCIENCE AND ENGINEERING

Linköping Studies in Science and Technology, Dissertation No. 2138, 2021
Department of Electrical Engineering

Linköping University
SE-581 83 Linköping, Sweden

www.liu.se