CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

MASTER'S THESIS

ISSN 1213-2365

All-speed Long-term Tracker Exploiting Blur

Denys Rozumnyi

rozumden@cmp.felk.cvut.cz

24 May 2019

Thesis Advisor: prof. Ing. Jiří Matas, Ph.D.

This work has been supported by the Czech Science Foundation grant GA18-05360S "Solving inverse problems for the analysis of fast moving objects".

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

All-speed Long-term Tracker Exploiting Blur

Denys Rozumnyi

24 May 2019

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Rozumnyi Denys** Personal ID number: **431566**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer Vision and Image Processing**

II. Master's thesis details

Master's thesis title in English:

All-speed Long-term Tracker Exploiting Blur

Master's thesis title in Czech:

Dlouhodobý tracker všech rychlostí s použitím rozmazání

Guidelines:

The problem of fast moving objects has been studied previously [1,2]. However, the proposed methods for tracking are limited by many assumptions, fast motion in all frames, linear motion and high contrast between the foreground and the background. In standard tracking approaches, blur is typically assumed to be a problem causing tracking failures. However, it provides information about motion direction in a single frame.
Design, implement and test all-speed long-term tracker that handles speed where the object is highly blurred, to slow motion, full occlusion and temporary invisibility when the object is out of the field of view. The tracker should make use of blur as an indicator of object motion. The thesis will explore whether blur can be used to robustify the all-speed tracker.

Bibliography / sources:

[1] Denys Rozumnyi et al. „The World of Fast Moving Objects". In CVPR 2017.
[2] Jan Kotera, Filip Šroubek „Motion Estimation and Deblurring of Fast Moving Objects". In ICIP 2018.

Name and workplace of master's thesis supervisor:

prof. Ing. Jiří Matas, Ph.D., Visual Recognition Group, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **13.02.2019** Deadline for master's thesis submission: **24.05.2019**

Assignment valid until: **30.09.2020**

prof. Ing. Jiří Matas, Ph.D.	doc. Ing. Tomáš Svoboda, Ph.D.	prof. Ing. Pavel Ripka, CSc.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

v Praze dne ....

podpis autora práce

Author statement for undergraduate thesis

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university thesis.

Prague, date ....

Signature

Abstract

Objects moving at high speeds along complex trajectories often appear in videos, especially videos of sports. Such objects move over non-negligible distances during exposure time of a single frame and therefore their position in the frame is not well defined. We propose a novel approach Tracking by Deblatting (TbD) which is based on the observation that motion blur is directly related to the intra-frame trajectory of an object. Blur is estimated by solving two intertwined inverse problems, blind deblurring and image matting, which we call deblatting. Non-causal TbD method estimates continuous, complete and accurate object trajectories. Full trajectory is estimated by fitting piecewise polynomials, which model physically justifiable trajectories. The output is a continuous trajectory function which assigns location for every real-valued time stamp from zero to the number of frames. As a result, tracked objects are precisely localised with higher temporal resolution than by conventional trackers. The proposed TbD tracker was evaluated on a newly created dataset of videos with ground truth obtained by a high-speed camera using a novel TIoU metric that generalises the traditional Intersection over Union and measures accuracy of intra-frame trajectories. Template learning in combination with a standard long-term tracker allows for long-term object tracking in all speeds. We show that from the trajectory function precise physical calculations are possible, such as radius, gravity or sub-frame object velocity. Results show high performance of TbD in terms of TIoU, recall and speed estimation.

Keywords: fast moving objects, deblurring, deblatting, tracking, trajectory estimation

Abstrakt

Objekty pohybující se vysokou rychlostí podél složitých trajektorií se často objevují ve videích, zejména ve sportovních videích. Takové objekty se během doby expozice jednoho snímku pohybují přes nezanedbatelné vzdálenosti, a proto jejich poloha v jednom snímku není přesně definována. Navrhujeme nový koncept Tracking by Deblatting (TbD), který je založen na pozorování, že rozmazání způsobené pohybem přímo souvisí s trajektorií objektu v rámci jednoho snímku. Rozmazání se odhaduje řešením dvou závislých inverzních problémů, "blind deblurring" a "image matting", které nazýváme "deblatting". Nekauzální TbD metoda odhaduje spojité, úplné a přesné trajektorie objektů. Celá trajektorie se nalezne pomocí napasování po částech spojitých polynomů, které modelují fyzicky věrohodné trajektorie. Výstupem je spojitá funkce, která přiřazuje polohu objektu pro každý časový okamžik daný reálným číslem od nuly do počtu snímků. Výsledkem je, že sledované objekty jsou přesně lokalizovány s vyšším časovým rozlišením než výstup standardních sledovacích metod. Navržená sledovací TbD metoda byla vyhodnocena na nově vytvořené datové sadě videí s anotacemi získanými vysokorychlostní kamerou s využitím nové metriky TIoU, která zobecňuje tradiční průnik nad sjednocením (IoU) a měří přesnost trajektorií v rámci jednoho snímku. Učení šablony v kombinaci s dobře fungující tradiční sledovací metodou umožňuje dlouhodobé sledování objektů libovolných rychlosti. Ukazujeme, že z funkce trajektorie jsou možné přesné fyzikální výpočty, jako je například výpočet poloměru, gravitace nebo rychlosti objektu v rámci jednoho snímku. Výsledky ukazují vysokou úspěšnost TbD z hlediska TIoU, pokrytí a přesnosti odhadu rychlosti.

Klíčová slova: rychle se pohybující objekty, deblurring, deblatting, tracking, nalezení trajektorie

Acknowledgements

First of all, I sincerely thank my supervisor Jiří Matas for his patience and professional guidance during the last five years. His inspiration, enthusiasm, motivation and fruitful discussions have kept me going since the beginning of my studies. He has taught and helped me more than I could ever give him credit for here.

I am indebted to my collaborators Filip Šroubek and Jan Kotera from UTIA who helped me with the project and implementation. Without their participation and contribution, this thesis could not have been successfully completed.

My academic career started at CMP and this accomplishment would have been impossible without my colleagues and friends from there. I am grateful to James Pritts for enlightening me the first glance of research. My thanks also goes to Jan Sláma for his valuable support throughout these years.

I would like to give special thanks to my family and the closest ones for their support, encouragement and understanding.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The field of visual object tracking has progressed significantly in recent years [WLY13, K+16, KML+16a, K+19]. The area covers a wide range of problems, including single object model-free short-term tracking [LVC+17, DHSKF14, VNM13, TYZW18] where a single target is localised in a video sequence given a single training example, assuming no occlusion or disappearance from the field of view, long-term tracking covering methods requiring re-detection and learning [KMM12, M+16b, MG17, T+17], multi-target multi-camera tracking [R+16, RT18], multi-view methods [KDVG14] and methods targeting specific objects, e.g. cars [B+00], humans [MD03] or animals [F+00]. Many variants of the problems have been considered – static or dynamic cameras or environments, RGBD input, use of inertial measurement units, to name a few. The interest to this field has been growing with Visual Object Tracking (VOT) challenges [KML+15, K+16, K+19, KML+16b, KML+16a] which started in 2013 and the seventh VOT 2019 challenge is being organised this year addressing short-term, long-term, real-time, RGB, RGBT and RGBD tracking.

Detection and tracking of fast moving objects is an underexplored area of tracking. In a paper focusing on tracking objects that move very fast with respect to the camera, Rozumnyi et al. [RKŠ+17, Roz17] presented the first algorithm that tracks such objects, i.e. objects that satisfy the Fast Moving Object (FMO) assumption – the object travels a distance larger than its size during exposure time. The authors have shown that the performance of standard state-of-the-art trackers drops significantly in the presence of FMOs, due to the effect of blur – the objects appear as semi-transparent streaks. Examples of applications with FMOs include tracking of balls and ball-like objects in sport videos, particles in scientific experiments, and flying birds and insects. However, the method proposed in [RKŠ+17] operates under restrictive conditions – the motion-blurred object should be visible in the difference image and trajectories in each frame should be approximately linear.

Standard trackers, both long and short term, usually provide information about the object location in a frame in the from of a single rectangle. This gives only one point of object location. In case if the output is a segmentation, then object location is even hardly defined. The true, continuous trajectory of the object centre is thus sampled with the frequency equal to the video frame rate. For slow moving objects, such sampling is adequate. For fast moving objects, especially if their trajectory is not linear (bounces, gravitation, friction), a single location estimate per frame cannot represent the true trajectory well, even if the fast moving object is inside the reported bounding box or segmentation. Moreover, standard trackers typically fail even in achieving that [RKŠ+17].

In the bachelor thesis [Roz17], Rozumnyi introduced a method for FMO detection and track-

Tracking by Deblatting | Non-causal trajectory estimation | Speed estimation

Figure 1.1. Trajectory reconstruction using the non-causal Tracking by Deblatting (middle) compared to the causal TbD (left). Colour codes trajectory accuracy, from red (complete failure) to green (high accuracy). The ground truth trajectory from a high-speed camera is shown in yellow, mostly under the estimated trajectory. Speed estimation output is visualised on the right. The ground truth speed (olive) is noisy due to discretisation and TbD speed estimates (lightgray) are inaccurate, which is fixed by the proposed TbD-NC (purple).

ing over a large range of speeds. However, the method is based on difference images which cannot successfully handle objects of very high speeds due to almost no contrast. On top of that, the method is not mathematically justifiable. We propose a novel methodology for tracking fast-moving, blurred objects. The approach untangles the image formation by solving two inverse problems: *motion deblurring* and *image matting*. We therefore call the method *Tracking by Deblatting*, TbD in short.

The deblatting procedure simultaneously recovers the trajectory of the object, its shape and appearance. We introduce a strong prior on the blur kernel and force it to lie on a 1D manifold. The corresponding curve models the object trajectory within a frame. Unlike a standard general tracker, TbD does not need a template of the object, since the representation of the shape and appearance of the object is recovered on the fly. Experiments show that the estimated trajectory is often highly accurate (see Figure 1.1).

We show that TbD performs well for both fast moving objects, slow moving objects and objects not moving at all. This makes TbD an *all-speed* method for object tracking. By combining TbD method with a state-of-the-art long-term tracker FuCoLoT [LČZV+18] and adding template learning to TbD, we make the method *long-term* for fast motion and low motion. TbD can thus successfully detect and track objects which speed up, slow down, disappear and appear as either fast moving or slow moving.

In its core, TbD assumes causal processing of video frames, i.e. the trajectory reported at the current frame is estimated using only information from previous frames. Applications of detection and tracking of fast moving objects do not usually require online and causal processing. FMOs move over distances so quickly that they could travel the scene twice in one second. Moreover, non-causal trajectory estimation brings many advantages, such as complete and accurate trajectories, which are among TbD limitations, e.g. failures at contact with a player or missing detections.

We also study non-causal Tracking by Deblatting (TbD-NC) and show that global analysis of FMOs leads to accurate estimates of FMO properties, such as nearly uninterrupted trajectory, velocity and shape. Figure 1.1 shows an example of non-causal trajectory estimation, which makes the trajectory more accurate and continuous throughout the entire sequence.

1.1. Contributions

Compared to the bachelor thesis [Roz17], the introduced method makes use of deblurring and fitting to estimate accurate and complete trajectories, which was entirely missing before. Thorough experiments on a new dataset with ground truth trajectories from a high-speed camera are

performed. We compare several variants of the TbD method and make an ablation study of the most important components. The thesis provides several important contributions over the FMO method [RKŠ+17] and the bachelor thesis [Roz17]. They are following:

- Novel approach to track objects in all-speed fashion is introduced. Objects can range from very fast and blurred objects as in [RKŠ+17] to standard moving objects or even objects with no motion. We show that Tracking by Deblatting can handle different motions. TbD is a long-term method which is able to learn object appearance and detect the object again when it is lost. TbD is based on solving two inverse problems of deblurring and image matting, followed by curve fitting. Previous approaches used only difference images and were not mathematically justifiable in contrast to TbD.
- We introduce a global non-causal TbD method, referred here as TbD-NC, for estimating *continuous* object trajectories by optimising a global criterion on the whole sequence. Segments without bounces are found by an algorithm based on dynamic programming, followed by robust fitting of polynomials using a least squares linear program. Recovered trajectories give the object location in every real-valued time stamp.
- Compared to the causal TbD, TbD-NC reduces by a factor of 10 the number of frames where the trajectory estimation by TbD completely fails.
- We show that TbD-NC increases the precision of the recovered trajectory to a level that allows good estimates of object velocity and size. Calculations of object radius, speed and gravitational force are shown. Experimental section confirms the accuracy of such estimates.
- Experiments are done on a newly created dataset with ground truth trajectories from a high-speed camera. Dataset and used data will be made publicly available at `http://cmp.felk.cvut.cz/fmo`.

Demo version of fast moving object detection is publicly available at `https://github.com/rozumden/fmo-cpp-demo` which is based on Aleš Hrabalík's implementation [Hra17]. Implementation of this thesis is available in the attached CD (see Appendix A) and online at `http://cmp.felk.cvut.cz/fmo`.

1.2. Thesis Structure

We discuss related work in Chapter 2. Then the posed problem and the solution, Tracking by Deblatting, are introduced in Chapter 3. In Chapter 4 we explain non-causal Tracking by Deblatting. Experiments on several datasets as well as applications are shown in Chapter 5. The TbD dataset is introduced in the experimental section. The thesis is concluded in Chapter 6.

CHAPTER 2

RELATED WORK

Object tracking methods are based on diverse principles, such as discriminative correlation filters [BCR15, DHSKF14, DHSKF15, LVC+17, TYZW18], feature point tracking [TK91], mean-shift [CRM03, VNM13], and tracking-by-detection [ZMS14, HGS+16]. In addition, several surveys of object tracking have been compiled [Avi07, BYB11, GRB13]. Excellent performance in visual object tracking has been shown by discriminative correlation filters [BCR15, DHSKF14, DHSKF15, LVC+17], yet all the methods fail when the tracked object is blurred as demonstrated in [RKŠ+17].

Recently, Lukežič et al. [LVC+17] proposed a new correlation-based tracker – CSR-DCF, which achieved state-of-the-art results on standard tracking datasets [K+16] and runs close to real-time on a CPU. The long-term version of CSR-DCF, the Fully Correlational Long-Term (FuCoLoT) tracker [LČZV+18], can even handle more difficult scenarios. The implementations of these methods are available online and therefore we use them as baseline methods for standard object tracking in the evaluation.

Methods proposed for object motion deblurring try to estimate sharp images from photos or videos without considering the tracking goal. Early methods worked with a transparency map (the alpha matte) caused by the blur, and assumed linear motion [Jia07, DW08] or rotation [SXJ07]. Blind deconvolution of the transparency map is better posed, since the latent sharp map is a binary image. Accurate estimation of the transparency map by alpha matting algorithms, such as [LLW08], is necessary and this is not tractable for large blurs. Other methods are based on the observation that autocorrelation increases in the direction of blur [KL14, SCXP15]. Autocorrelation techniques require a relatively large neighbourhood to estimate blur parameters and such methods are not suitable for small moving objects. More recently, deep learning has been applied to motion deblurring of videos [W+17, S+17b] and to the generation of intermediate short-exposure frames [J+18]. The proposed convolutional neural networks are trained only on small blurs. Blur parameters are not available as they are not directly estimated.

Tracking methods that consider motion blur have been proposed in [W+11, S+17a, M+16a], yet there is an important distinction between models therein and the FMO problem considered here. The blur is assumed to be caused by camera motion and not by the object motion, which results in blur affecting the whole image and in the absence of alpha blending of the tracked object with the background.

The problem we are interested in can be viewed as an alpha matting of the background and blurred object of interest. In order to always have non-zero influence of the background, we consider fast moving objects that move over a distance larger than their size in one exposure

time. The goal is to create a method which handles fast moving objects as well as standard moving objects which move over a distance lower than their size, thus they fully occlude the background in some regions.

To our knowledge, the only published method that tackles the similar problem of tracking motion-blurred objects remains the work in [RKŠ+17]. The authors assume linear motion and the trajectories are calculated by morphological thinning of difference image between the given frame and the estimated background. Deblurring of fast moving objects has also appeared recently in a work by Kotera et al. [KŠ18], but they do not consider FMO tracking or detection.

There are two improvements over the work in [RKŠ+17]: the Master's thesis of Aleš Hrabalík [Hra17], which is focused on real-time implementation of FMO detector, and the Bachelor thesis of Denys Rozumnyi [Roz17], which improves the precision and recall of the FMO detector, but both methods are still based on the difference image. They also lack mathematical background and have many limitations. In further experiments, we use the improved version (from the Bachelor thesis [Roz17]) of the pioneering work [RKŠ+17] when this work is referred to.

CHAPTER 3

CAUSAL TRACKING BY DEBLATTING

Tracking by Deblatting is a novel framework which unites deblurring, matting, tracking and long-term object trajectory estimation. In the following sections we will discuss each step in details.

The proposed method formulates tracking as an inverse problem to the video formation model. Suppose that within a single video frame I an object F moves along the trajectory $\mathcal{C}$ in front of background B. Frame I is then formed as

$$I = H * F + (1 - H * M)B, \tag{3.1}$$

where $*$ denotes convolution, H is blur kernel or the Point Spread Function (PSF) of the object motion blur corresponding to trajectory $\mathcal{C}$, and M is the binary mask of the object shape, i.e. the indicator function of F. We refer to the pair (F, M) as the object model. The first term in the formation model (3.1) is the tracked object blurred by its own motion, the second term is the background partially occluded by the object, and the blending coefficients are determined by $H * M$. Inference under the assumption of this formation model consists of solving simultaneously two inverse problems: blind deblurring and image matting. The solution is the estimated blur kernel H and the object model F and M.

Motion blur in (3.1) is modelled by convolution which implies the following assumption about the object motion: The object shape and appearance remain constant during the frame exposure time. Scenarios that satisfy the assumption precisely are following. Either an object of arbitrary shape is undergoing only translational motion or a spherical object of uniform colour undergoing arbitrary motion under spatially-uniform illumination. In addition, the motion must be in a plane parallel to the camera image plane to guarantee constant size of the object. For the purpose of tracking and trajectory estimation we claim that the formation model (3.1) with convolution is sufficient as long as the assumption holds at least approximately, which is experimentally validated on the presented dataset which contains rotating objects of various shapes and colourings.

The proposed TbD method is iterative and causal processing of a new frame I_{i+1} using only knowledge acquired from earlier frames $\{I_1, \ldots, I_i\}$. Figure 3.1 (shaded area) provides an overview of the entire TbD pipeline. Inputs are the current estimates of the object model F_i and M_i, the background B_i, and a region of interest (ROI) D_i in I_{i+1}, which is the neighbourhood of the predicted object location. Outputs are object model F'_{i+1} and M'_{i+1} which are used for updating the model, estimated blur kernel H'_{i+1} and the final curve $\mathcal{C}'_{i+1}$ computed from the blur kernel. All accumulated curves $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$ and the corresponding blur kernels are outputs of TbD.

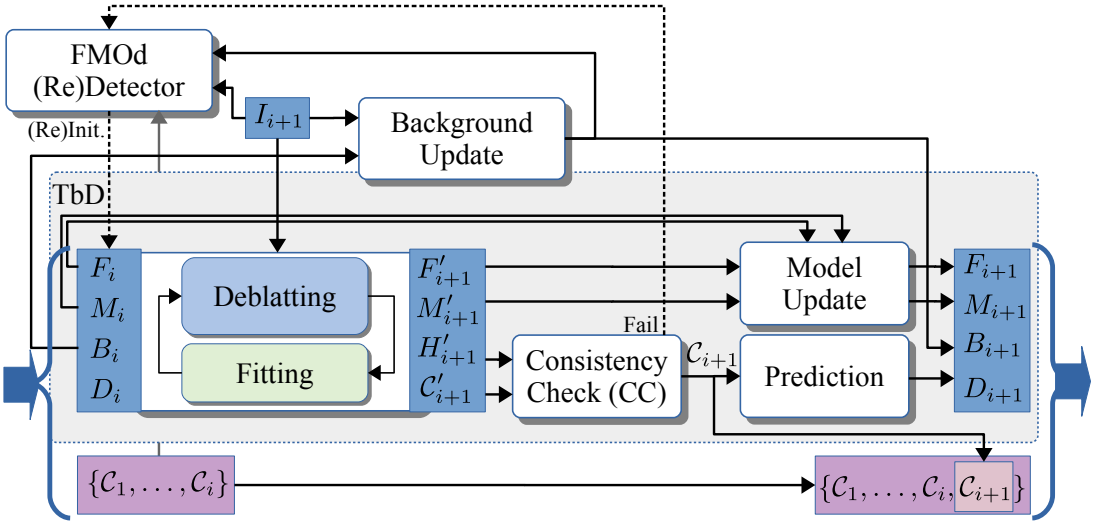


Figure 3.1. Long-term All-speed Tracking by Deblatting. TbD sequentially processes video frames $\{I_i\}$ and estimates trajectory curves $\{\mathcal{C}_i\}$ of the tracked object F_0. Iterative deblatting and trajectory fitting generates new estimates of the object model (appearance F and shape M) and blur H with the trajectory fit $\mathcal{C}$. If the blur and trajectory pass a consistency check, extrapolation of the trajectory predicts the region of interest D in the next frame and both the object model and background B are updated. The FMO detector is activated during initialisation or if the consistency check fails.

Three main steps are performed in TbD:

1. *Deblatting*: Iteratively solve blind deblurring and matting in the image region D_i with the model (3.1) and estimate F'_{i+1}, M'_{i+1}, and H_{i+1}; see Section 3.1.

2. *Trajectory fitting*: Estimate physically plausible motion trajectory (parametric curve) $\mathcal{C}_{i+1}$ corresponding to H_{i+1} and optionally adjust D_i according to $\mathcal{C}_{i+1}$; see Section 3.2.

3. *Consistency check & model update*: Verify that the error of the mapping $H \to \mathcal{C}$ is below threshold τ, predict the new region of interest D_{i+1} for the next frame, and update the object model to F_{i+1} and M_{i+1}.

A more detailed illustration of Steps 1 and 2 is in Figure 3.2. Step 1 stops after reaching either a given relative tolerance or a maximum number of iterations. Steps 1 and 2 are repeated only if the newly fitted $\mathcal{C}$ touches the boundary of D – in this case the new D is the d-neighbourhood of $\mathcal{C}$ where d is the object diameter. Adjusting D this way helps to eliminate the detrimental influence of other moving objects to correct estimation of H.

If the consistency check (CC) passes, we extrapolate the estimated trajectory to the next frame and D_{i+1} is again d-neighbourhood of this extrapolation. To update the appearance model we use exponential forgetting

$$F_{i+1} = \gamma F_i + (1 - \gamma) F'_{i+1}, \tag{3.2}$$

where γ is a real number between zero and one. M is updated analogically.

To enable long-term tracking, the FMO detector (FMOd) from [RKŠ+17] determines the new input if CC fails. First, FMOd tries detecting the object in a gradually enlarged D. The new proposal of object location is again validated by the three steps with template learned from previous frames. If it succeeds, the main TbD pipeline is reinitialised with D set as a neighbourhood of the FMOd detection. If FMOd fails, TbD returns the extrapolation of trajectory $\mathcal{C}_i$ as the best guess of $\mathcal{C}_{i+1}$ and tracking is restarted anew on the next frame. In case

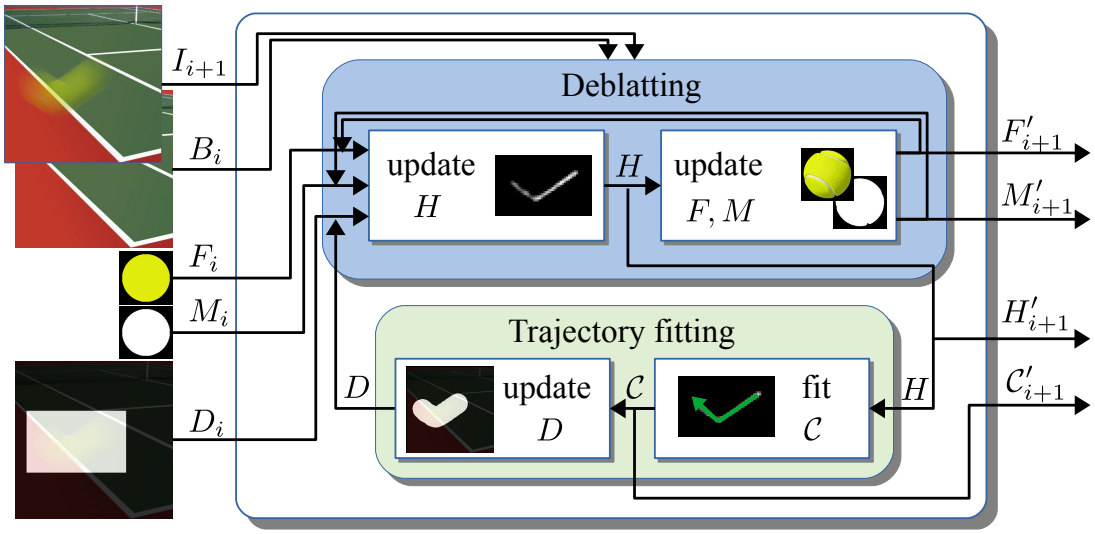


Figure 3.2. Deblatting with trajectory fitting. *Deblatting*, which consists of *debl*urring and m*atting*, is described in Section 3.1. After deblurring and matting, an attempt is made to fit the kernel H, described in Section 3.2. *Inputs:* incoming frame I_{i+1}, and current estimates of the object appearance F_i, shape M_i, background B_i and predicted region of interest D_i in I_{i+1}. *Outputs:* new estimates of the object appearance F'_{i+1}, shape M'_{i+1} and blur H'_{i+1} with the trajectory C'_{i+1}.

that object speed is lower than a given threshold, instead of trajectory extrapolation we initialise state-of-the-art long-term tracker FuCoLoT [LČZV+18] and use its output as the best guess. This enables long-term tracking even at low speeds, when deblatting does not perform so well and where a lot of research has been done to create well-performing trackers.

The background B_i is estimated as a temporal median of frames B_{i-1}, B_{i-2}, ..., optionally including video stabilisation if necessary. The first detection is also performed automatically by FMOd. The object appearance model is either learned "on the fly" starting trivially with $F_0 \equiv 1$, $M_0 \equiv 1$, which we call TbD-T0. Alternatively, the user provides a template of the tracked object, e.g. a rectangular region from one of the frames where the object is still. This version is denoted by TbD-T1.

Deblatting works not only for fast motion, but also for low to zero motion. In case of an object which stays still, the blur kernel H would contain only a single point and fitting is trivial. If the object abruptly becomes fast moving, e.g. somebody hits the object, then the prediction step will usually fail and the method waits for the next FMO detection. This implies that the proposed method is an all-speed tracker.

Long-term for fast motion is achieved by applying deblatting to FMOd with reconstructed object appearance from previous frames as a template. The recent state-of-the-art Fully Convolution Long-term Tracker (FuCoLoT) [LČZV+18] makes the method long-term for low motion.

So far, only weak relation exists between trajectories in adjacent frames and there is no hard constraint that the trajectory in previous frame must be consistent with the trajectory in the following frame. Also due to partial exposure, we always have a gap between consequent trajectories. Only in the ideal case of the full exposure, they could potentially form continuous trajectories. But in most cases, the last point in previous frame does not equal the first point in the following frame. Applying such a hard constraint in deblatting will limit its efficiency and will require difficult combinatorial problem of simultaneous deblatting in all frames together.

We relax the continuity in sequence constraint during deblatting and construct continuous trajectory through the whole sequence as a post-processing problem, where the trajectory is estimated by dynamic programming, followed by fitting polynomial functions which explain object motion. This final version of TbD is called non-causal Tracking by Deblatting (TbD-NC)

Input I | High FPS | Blur H | F | M

Figure 3.3. Deblatting examples – shuttlecock (top) and volleyball (bottom). From left to right: the input image, the corresponding high-speed camera frame, the estimated blur H, the estimated appearance F and the shape M.

and it is explained in Chapter 4.

3.1. Deblatting

The core step of TbD is the extraction of motion information H from the input frame, which we formulate as a blind deblurring and matting problem. Inputs are the frame I, domain D, background B, and the object appearance model $\hat{F}$. The inverse problem corresponding to (3.1) is formulated as

$$\begin{aligned} &\min_{F,M,H} \frac{1}{2} \|H * F + (1 - H * M)B - I\|_2^2 \\ &+\frac{\lambda}{2}\|F - M\hat{F}\|_2^2 + \alpha_F\|\nabla F\|_1 + \alpha_H\|H\|_1 \end{aligned} \tag{3.3}$$

s.t. $0 \leq F \leq M \leq 1$ and $H \geq 0$ in D, $H \equiv 0$ elsewhere. The primary unknown is H, but F and M are estimated as by-products. The first term in (3.3) is the fidelity to the model (3.1). The second λ-weighted term is a form of "template-matching", an agreement with a prescribed appearance. The template $\hat{F}$ is multiplied by M because if $\hat{F}$ is initially supplied by user as a rectangular region from a video frame, it contains the object and partially also the surrounding background.

When processing the i-th frame, we set $\hat{F} = F_{i-1}$ as the updated appearance estimate (3.2) from the previous frame. The first L^1 term is the total variation that promotes smoothness of the recovered object appearance. The second L^1 regularisation enforces sparsity of the blur and reduces small non-zero values.

If M is a binary mask then the condition $F \leq M$ states that F cannot be non-zero where M is zero – pixels outside the object must be zero. For computational reasons, we relax the binary restriction and allow M to attain values in the range $[0, 1]$. The correct constraint corresponding to this relaxation is then exactly $F \leq M$, assuming F alone is bounded in $[0, 1]$. Relaxing the binary constraint also makes it easier to update the model with exponential forgetting factor (3.2), as γ value is usually a floating point number.

The inequality constraint $H \geq 0$ prohibits negative values in H, which are physically implausible for motion blur. For computational speed-up, H is estimated only within the do-

main D.

We solve (3.3) in an alternating manner, fix (F, M) and solve for H and vice versa, until convergence.

Minimising (3.3) with respect to H with (F, M) fixed becomes

$$\min_H \frac{1}{2} \|H * F + (1 - H * M)B - I\|_2^2 + \alpha_H \|H\|_1 \tag{3.4}$$

s.t. $H \geq 0$. We use Alternating Direction Method of Multipliers (ADMM) to solve (3.4).

Minimising (3.3) with respect to the joint unknown (F, M) with H fixed is

$$\min_{F,M} \frac{1}{2} \|H * F + (1 - H * M)B - I\|_2^2 + \frac{\lambda}{2} \|F - M\hat{F}\|_2^2 + \alpha_F \|\nabla F\|_1 \tag{3.5}$$

s.t. $0 \leq F \leq M \leq 1$. We solve this problem using again ADMM[1].

To summarise, the alternating H–(F, M) estimation loop for the i-th frame proceeds as follows:

1. Initialise $M := M^{i-1}$ (if available from previous detection) or $M \equiv 1$; initialise $\hat{F} := F^{i-1}$, $F := M\hat{F}$.
2. Calculate H by solving (3.4).
3. Check convergence, exit if satisfied.
4. Calculate (F, M) by solving (3.5), go to 2.

Examples of the deblatting alone are in Figures 3.3 and 3.4. Figure 3.3 contains from left to right the input frame (crop), corresponding frame from the high-speed camera, estimated blur kernel H, estimated object F and object shape M. In the top row, we see that the shape of the badminton shuttlecock, though not circular, is estimated correctly. In the bottom row, we see that if the non-uniform object undergoes only small rotation during motion, the appearance estimation can also be good. In this case, the shape estimation is difficult due to the mostly homogeneous background similar to the object.

Figure 3.4 is another interesting example of the deblatting behaviour. The input frame is in the top left corner and the corresponding part from the high-speed camera is below. The object casts significant shadow. If we set the size of F too small, the model cannot cope with the shadow and the estimated blur will contain artefacts in the locations of the shadow as is visible in the top row. If instead we make the support of F sufficiently large, the estimated mask compensates for the shadow and the estimated blur is clean as shown in the bottom row. It also means that F does not only represent the object itself, but it can also explain some other phenomena in the region of interest or it can even represent image noise.

3.2. Trajectory Fitting in Frame

Fitting the blur kernel H, which is a grey-scale image, with a trajectory $\mathcal{C}(t) : [0, 1] \to \mathbb{R}^2$ serves three purposes. First, we use the error of the fit in the Consistency Check to determine if H is the motion blur induced by the tracked object and thus whether to proceed with tracking, or to declare the deblatting step a failure and to reinitialise it with different parameters. Second, the trajectory as an analytic curve can be used for motion prediction whereas H cannot. Third, $\mathcal{C}$ defines the intra-frame motion, which is the desired output of the proposed method.

[1] Implementation of ADMM is kindly provided by collaborators.

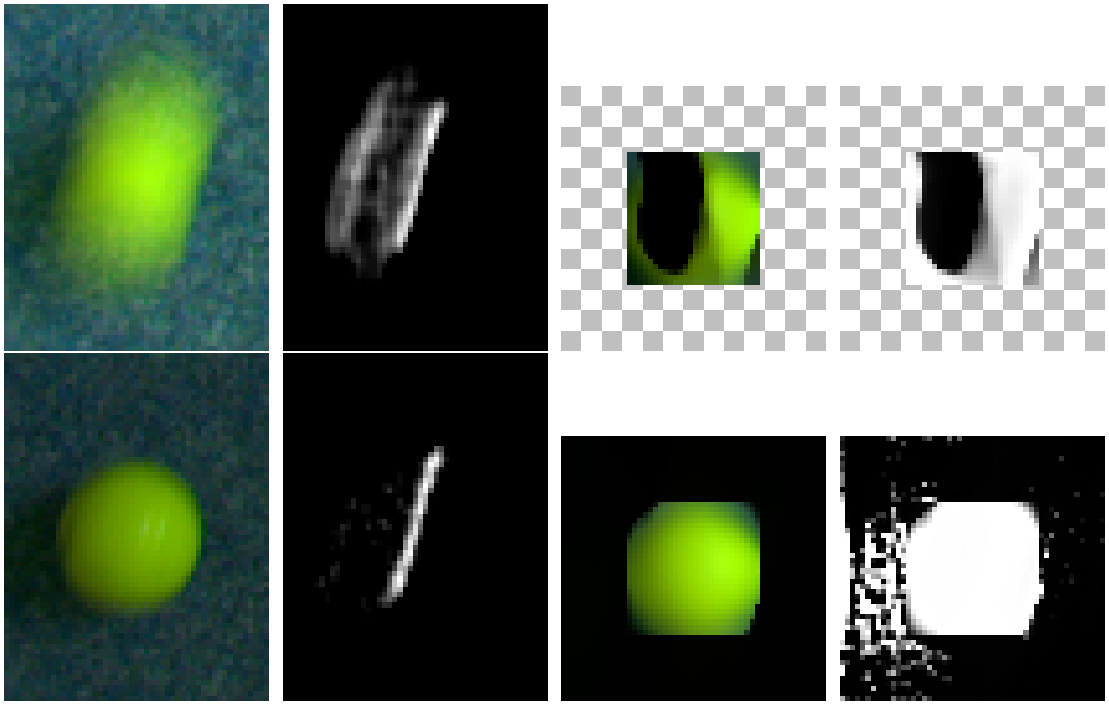


Figure 3.4. The shadow and blur estimation. *Top:* the domain of F is set too small and the shadow causes artefacts in H. *Bottom:* the domain of F is larger, M can compensate for the shadow and the blur H is estimated correctly.

The goal of TbD is to produce a precise intra-frame motion trajectory, and not only a single position per frame in the form of a bounding box.

The fitting is analogous to vectorisation of raster images. It is formulated as the maximum a posteriori estimation of $\mathcal{C}$, given H, with the physical plausibility of the trajectory used as a prior. Let $\mathcal{C}$ be a curve defined by a set of parameters θ (e.g. polynomial coefficients) and $H_\mathcal{C}$ be a raster image of the corresponding $\mathcal{C}$ (i.e. blur PSF). We say that the curve $\mathcal{C}$ is the *trajectory fit* of H if θ minimises

$$\min_\theta \|H_\mathcal{C} - H\| \quad \text{s.t. } \mathcal{C} \in \mathcal{M}, \tag{3.6}$$

where $\mathcal{M}$ is the set of admissible curves.

Our main tracking targets are balls and similar free-falling objects, therefore the assumption is that between impulses from other moving objects (e.g. players), tracked objects can be approximated in one frame as objects in free flight or objects which bounce off static rigid bodies. We then define $\mathcal{M}$ as a set of piecewise quadratic continuous curves – quadratic to account for de-acceleration due to gravity and piecewise to account for abrupt change of motion during bounces. $\mathcal{C} \in \mathcal{M}$ is defined as

$$\mathcal{C}(t) = \begin{cases} \sum_k^2 c_{k,1} t^k & 0 \leq t \leq \tilde{t}, \\ \sum_k^2 c_{k,2} t^k & \tilde{t} \leq t \leq 1, \end{cases} \tag{3.7}$$

s.t. $\sum_k^2 c_{k,1} \tilde{t}^k = \sum_k^2 c_{k,2} \tilde{t}^k$. Single linear or quadratic curves are included as special cases when $\tilde{t} = 1$.

Let us view the blur H as a set of pixels with coordinates x_i and intensities $w_i > 0$. Sequential RANSAC finds line segments as follows: sample two points, find inliers of the corresponding line, find the most salient consecutive run of points on this line and in each round remove

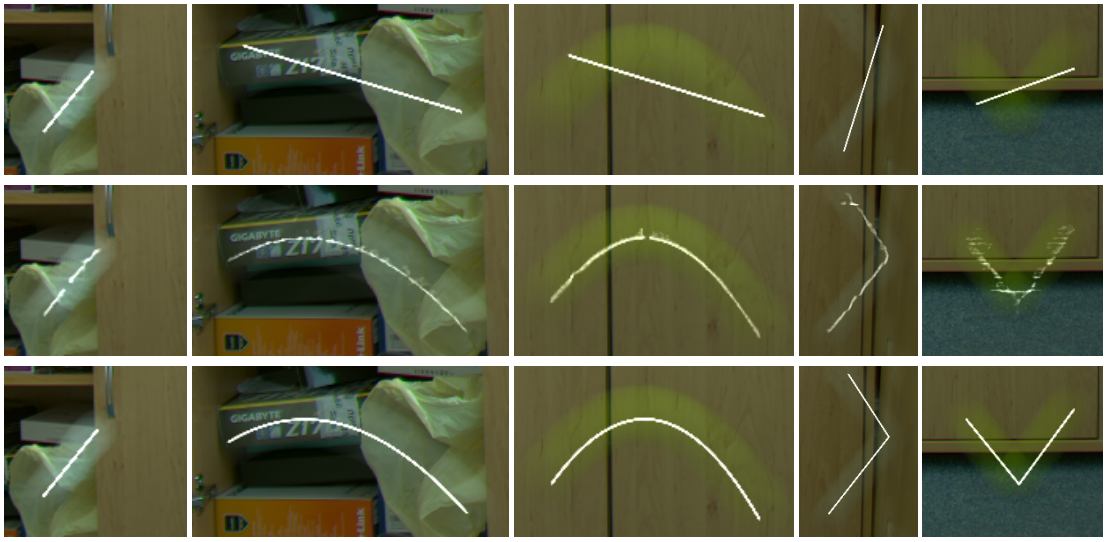


Figure 3.5. Intra-frame trajectory estimation by Tracking by Deblatting. Close-ups of the tracked object. Superimposed in white: trajectory estimated by the FMO detector (top row), blur kernel H estimated by TbD (middle row) and the final trajectory returned by TbD (bottom row). Examples of (left to right) a linear motion, curved motions and bounces.

the winner from the sampling pool. The saliency is defined as $\sum w_i$ for x_i in the inlier set and "consecutive" means that the distance between neighbouring points is bounded by a threshold. The search stops when the saliency drops below a specified threshold or there are no more points. We denote the set of collected linear segments as $\mathcal{M}_1$. Parabolic arcs are found similarly. We sample four points, find two corresponding parabolas, project the remaining points on the parabolas to determine the distance and inlier set as well as the arc-length parametrisation of inliers (required for correct ordering and mutual distance calculation of inliers) and again find the most salient consecutive run. We denote the set of collected parabolic segments as $\mathcal{M}_2$.

The solution will be in the vicinity of a curve formed from one or two segments (linear or parabolic) found so far. Let $\mathcal{C}_1, \mathcal{C}_2 \in \mathcal{M}_1$ be two linear segments. If the intersection P of the corresponding lines is close to the segments (with respect to some threshold), the curve connecting $\mathcal{C}_1 \rightarrow P \rightarrow \mathcal{C}_2$ is a candidate for the piecewise linear trajectory fit. This way we construct a set $\mathcal{M}_3$ of all candidate and similarly $\mathcal{M}_4$ with candidates of parabolic pairs.

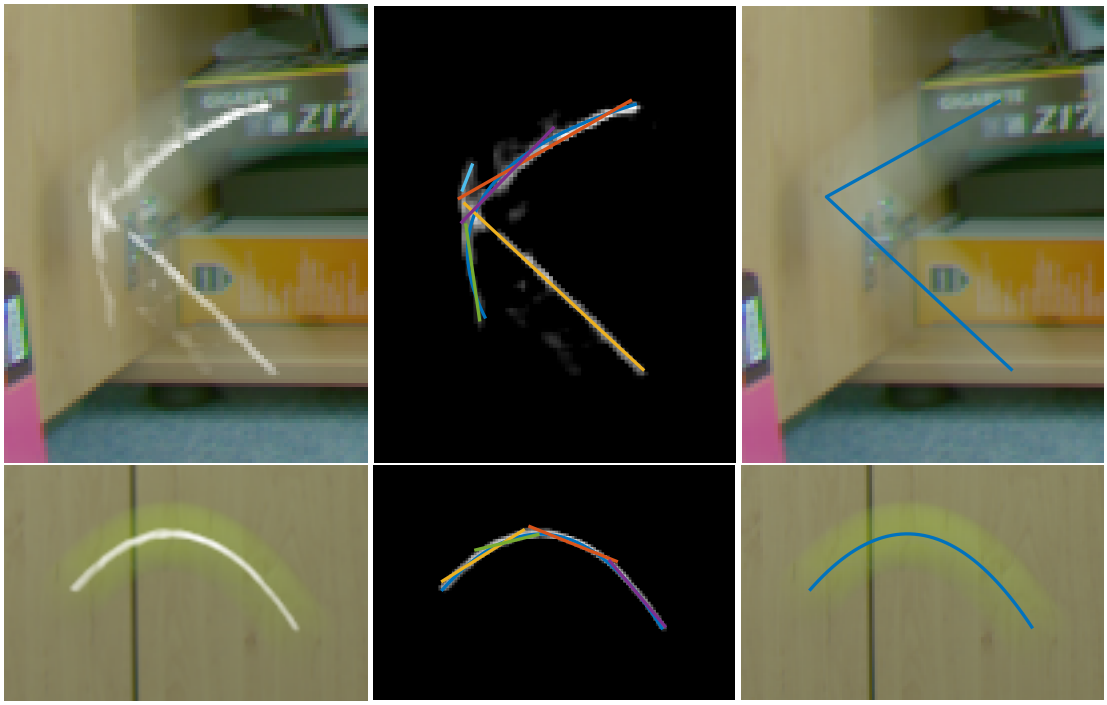
Finally, for each curve $\mathcal{C} \in \mathcal{M} = \bigcup \mathcal{M}_i$ we construct $H_\mathcal{C}$, measure the error $\|H_\mathcal{C} - H\|$ and choose the best candidate as the trajectory fit.

In TbD, the Consistency Check of the trajectory fit $\mathcal{C}$ is performed by evaluating the criterion

$$\frac{\|H_\mathcal{C} - H\|}{\|H\|} < \tau. \tag{3.8}$$

Figure 3.6 shows examples of trajectory estimation. The left column is the input image with the estimated PSF superimposed in white and the right column shows the estimated motion trajectory. The efficacy of trajectory fitting is a crucial part of the framework, the estimated blur can contain various artefacts (e.g. in the top example due to the ball shadow) and the trajectory fit still recovers the actual motion.

One of the benefits of TbD is its ability to produce a precise intra-frame motion trajectory. Most trackers provide output in the form of a bounding box, FMOd outputs line segments; the deblurring loop of the TbD provides richer trajectory information. Figure 3.5 presents several examples. The top row contains close-ups of the tracked object in the input frame with



I and H RANSAC I and $\mathcal{C}$

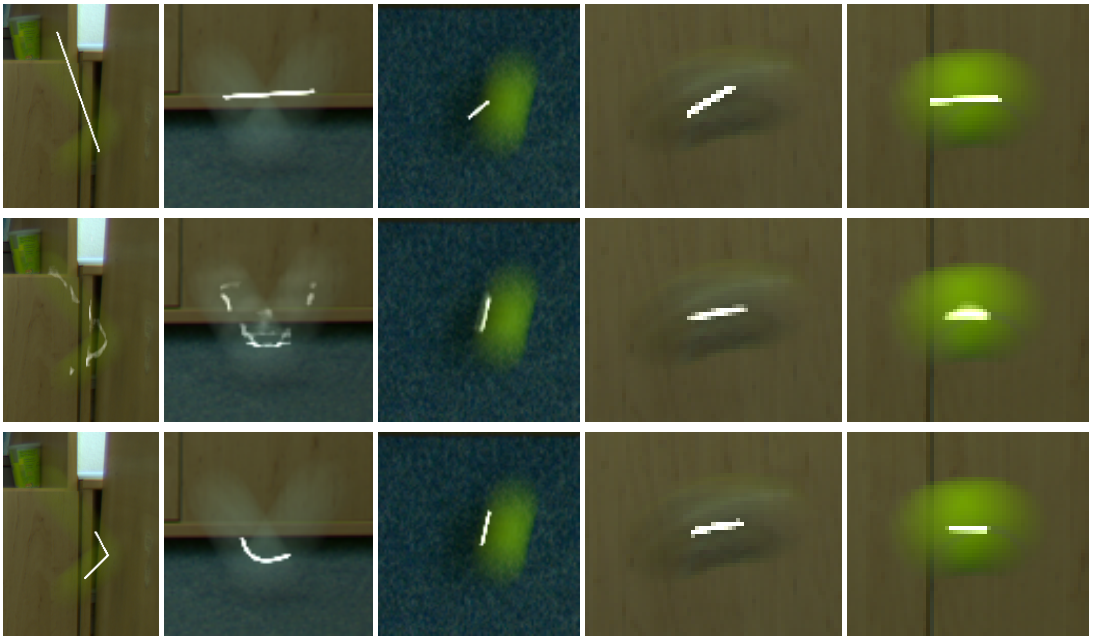
Figure 3.6. Trajectory fitting in one frame. Input image with estimated blur superimposed in white (left), linear and parabolic segments found by RANSAC (middle), final fitted trajectory (right).

superimposed line fit provided by FMOd in white. The second row shows the blur H estimated by TbD and the third row shows the final trajectory returned by TbD after curve fitting. The examples are ordered, left-to-right, from simple to complicated. FMOd copes well when the trajectory is linear but fails to provide accurate output in other cases – parabolic trajectory or when the direction of motion changes during exposure. TbD works well even in these cases. The examples show that the trajectory fitting step is a crucial part of the framework, in some cases the estimated blur is noisy – broken into several pieces or containing various artefacts – and the trajectory fit recovers the actual motion.

Figure 3.7 has a similar structure but provides examples worth attention and failure cases. A frequent problem in the deblurring phase is caused by background changes during exposure, e.g. due to shadows cast by the object or when the object bounces off a non-stationary object. In this case, the estimated blur contains artefacts not related to the object motion but rather compensating the background change. The artefacts may cause a failure of the trajectory fitting as shown in Figure 3.7 (a) and (b). In Figure 3.7 (c), the shadow "moves" with the object and is tracked as though it were part of the object. This causes that the estimated trajectory is shifted. In some cases, especially when there is low contrast between the object and the background, the trajectory is clipped, as in Figure 3.7d.

3.3. Motion Prediction

Performing deblurring on the whole input frame is not feasible. Deblurring is rather slow, but more importantly, the video frame typically contains other objects in motion and those can cause problems in discerning the motion of the tracked object. For this reason, we calculate blur kernel H and object model F only in a selected region of interest (ROI), where the tracked object is most likely to appear. Given the motion trajectory from the last available step, we



(a) (b) (c) (d) (e)

Figure 3.7. Inaccurate intra-frame trajectory estimation by Tracking by Deblatting. Close-ups of the tracked object. Superimposed in white: trajectory estimated by FMOd (top row), blur kernel H estimated by TbD (middle row) and the final trajectory returned by TbD (bottom row). **(a)** - **(b)** blur H contains artefacts due to motion in the background and the shadow of the object causing inaccurate trajectory fitting, **(c)** shifted trajectory as the shadow is considered a part of the object and the trajectory is placed at the centre of this "virtual" object, **(d)** trajectory is too short due to poor contrast between the object and background, **(e)** the object is slow and trajectory fitting is less stable.

extrapolate the motion into the next frame with an assumption that the acceleration (or the velocity for linear curves) remains approximately constant between two consecutive frames. At the beginning of tracking (or after reinitialisation) when the direction of motion is unknown, two ROIs are considered by extrapolating the trajectory in both directions. Then the direction which gives a trajectory with higher consistency check will be chosen. The trajectory estimated by the FMO detector has no direction. It will be added one frame later when TbD prediction is made. If TbD does not succeed in the following frame, the trajectory reported by the FMO detector will stay without orientation.

Predictions are done by taking values of function $\mathcal{C}_t$ in the range of either $[1, 2] \subset \mathbb{R}$ or $[-1, 0] \subset \mathbb{R}$, depending on the orientation. To account for unexpected speed up, we extend these intervals by 0.5 in the direction of motion.

A new initialisation by FMOd is required when the motion prediction step fails and predicts incorrect region of interest D for the deblurring step. Motion prediction is prone to fail in the case of abrupt motion changes (bounces, accelerations) and when motion is slow (motion direction is ambiguous). Overestimating the ROI can solve the problem but increases the running time and probability of including other moving objects in the ROI. Having tested different variations of the proposed approach, we concluded that small ROIs with FMOd re-initialisation is more reliable.

Figure 3.8 shows how predictions from one frame to another are made. The shaded area, in which all computations are made, is updated in each iteration and in every frame. This speeds up the computations and also removes the influence of other moving objects. If the prediction is completely wrong, TbD waits for the next detection by the FMO detector and outputs only the prediction without deblatting, fitting and consistency check.

Figure 3.8. Examples of predictions in the TbD framework. From left to right: previous frame with estimated trajectory, current frame with predicted trajectory, estimated blur kernels, final trajectory fit in one frame. Predicted area in which computations are done is highlighted. Predictions are coloured in red. Current estimation of the trajectory is marked in a range from yellow to green, depending on the Trajectory-IoU.

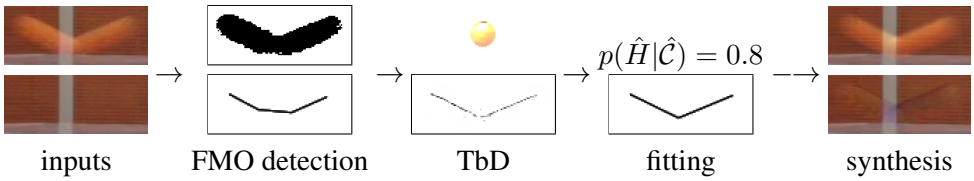


Figure 3.9. TbD framework accepting a true positive detection: FMO detection in the initialisation step detects a fast moving object and makes a rough guess of its trajectory. Blind deblurring in the E-step estimates the object appearance $\hat{F}$ and corresponding blur kernel $\hat{H}$. Trajectory fitting (M-step) $\hat{\mathcal{C}}$ is performed on $\hat{H}$ and the goodness of the fit is calculated as the conditional probability $p(\hat{H}|\hat{\mathcal{C}})$, which in this case is high. The image and background synthesis is not part of the TbD framework. It illustrates the accuracy of generating input data from estimated variables.

3.4. Maximum Likelihood Explanation

The idea of Tracking by Deblatting is to detect moving objects by estimating their physically plausible trajectory using a deconvolution algorithm. We first described the proposed framework and then discussed individual steps in detail. Now we will show another mathematical view on Tracking by Deblatting. Trajectory estimation, which is done by TbD in one observed frame I, is formally equivalent to a maximum likelihood problem

$$\hat{\mathcal{C}} = \arg\max_{\mathcal{C}} \ln p(I|\mathcal{C})\,. \tag{3.9}$$

The analytic expression of likelihood $p(I|\mathcal{C})$ is possible if latent variables, such as object F and blur kernel H, are introduced. Noting that $\mathcal{C}$ and H are dependent while latent F and H are independent, the likelihood with latent variables takes the form

$$p(I, \{F, H\}|\mathcal{C}) = p(I|F, H)p(F)p(H|\mathcal{C})p(\mathcal{C})\,. \tag{3.10}$$

Distributions on the right-hand side have analytic expression. The likelihood $p(I|F, H)$ is given by the noise distribution of N and substitution from the acquisition model (3.1). The object appearance prior $p(F)$ enforces the smoothness constraint of the object model $F(x)$. The trajectory prior $p(\mathcal{C})$ enforces the motion model (4.1) and $p(H|\mathcal{C})$ is the conditional distribution of the blur kernel H given the trajectory $\mathcal{C}$.

Marginalising $p(I, \{F, H\}|\mathcal{C})$ with respect to the latent variables $\{F, H\}$ is intractable and we therefore apply a variation of Expectation Maximisation (EM) algorithm. The expectation (E) step becomes

$$E_{\{F,H\}|I}[\ln p(I, \{F, H\}|\mathcal{C})] = \max_{F,H} \ln p(I, \{F, H\}|\mathcal{C})\,, \tag{3.11}$$

where $E_{\{F,H\}|I}[\cdot]$ denotes the expected value with respect to the conditional distribution of latent variables. To compute the E step effectively, we choose the conditional distributions of latent variables to be delta distributions and then the expected value is equal to the maximum value, which explains the equality in (3.11). The E step is thus similar to blind deconvolution, in which we solve iteratively an inverse problem associated with the formation model (3.1). The maximisation (M) step becomes

$$\hat{\mathcal{C}} = \arg\max_{\mathcal{C}} E_{\{F,H\}|I}[\ln p(I, \{F, H\}|\mathcal{C})] = \arg\max_{\mathcal{C}} p(\hat{H}|\mathcal{C})p(\mathcal{C})\,, \tag{3.12}$$

where $\hat{H}$ is the estimated blur kernel in the E step. The second equality follows from (3.10) and the M step is similar to a curve fitting problem.

As EM algorithms are prone to local maxima, a good initialisation is important, which is done by the FMO detection or prediction from the previous frame. Depending on the amount of prior

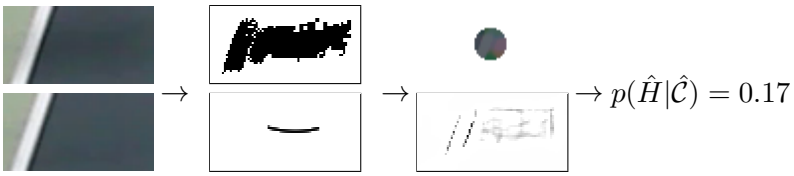


Figure 3.10. TbD framework rejecting a false positive detection caused by shadows: FMO detector makes a false detection. Blind deblurring in the EM-step estimates the most likely appearance of an object $\hat{F}$ and blur kernel $\hat{H}$ that would generate the input region. Since the region is not a result of the image formation model with convolution, $\hat{H}$ differs from any admissible trajectory and the probability $p(\hat{H}|\hat{\mathcal{C}})$ is thus low.

knowledge of the tracked object that is built on detection in previous frames, the prediction and FMO detector generates multiple candidates for trajectories. Trajectories are rendered as initial blur kernels H_0 and then validated by the EM steps.

The E step in the blind deblurring loop improves a blur estimate $\hat{H}$ and the M step returns a trajectory estimate $\hat{\mathcal{C}}$. The object detection is accepted or rejected based on the consistency check (3.8) between the estimated trajectory and blur kernel

$$p(\hat{H}|\hat{\mathcal{C}}) = \frac{\|\hat{H}_{\hat{\mathcal{C}}} - \hat{H}\|}{\|\hat{H}\|}\,. \tag{3.13}$$

We set the threshold for detection to 0.5, which was experimentally validated to be sufficient for separating false positives. Figure 3.9 shows an example of true positive detection by the TbD pipeline: initial inaccurate trajectory from FMO detection, improvement in the EM-step, and final curve fitting with probability calculation. An example of false positive detection by the FMO detection and final rejection in the EM-step is illustrated in Figure 3.10. FMO detector was upgraded with the proposed fitting approach rather than simple linear curve fitting as in [RKŠ+17].

CHAPTER 4
NON-CAUSAL TRACKING BY DEBLATTING

The outputs of the causal TbD are individual trajectories $\mathcal{C}_t$ and blur kernels H_t in every frame. They serve as inputs to the non-causal Tracking by Deblatting, which is based on post-processing of individual trajectories from Tracking by Deblatting. The final output of TbD-NC consists of a single trajectory $\mathcal{C}_f(t) : [0, N] \subset \mathbb{R} \rightarrow \mathbb{R}^2$, where N is a number of frames in the given sequence. The function $\mathcal{C}_f(t)$ outputs precise object location for any real number between zero and N. Each frame has unit duration and the object in each frame is visible only for duration of exposure fraction $\epsilon \leq 1$. Function $\mathcal{C}_f(t)$ is continuous and piecewise polynomial

$$\mathcal{C}_f(t) = \sum_{k=0}^{d_s} c_{k,s} t^k \quad t \in [t_{s-1}, t_s], s = 1..S, \tag{4.1}$$

with S polynomials, where polynomial c_s has degree d_s. The degree depends on the size of time-frame in which the polynomial c_s is fitted to. Variables t_s form splitting of the whole interval between 0 and N, i.e. that $0 = t_0 < t_1 < ... < t_{S-1} < t_S = N$.

Polynomials of degree 2 (parabolic functions) can model only free falling objects under the gravitational force. In many cases forces, such as air resistance or wind, also influence the object. They are difficult to model mathematically by additional terms. Furthermore, we would like to keep the function linear with respect to the weights. Taylor expansion will lead to a polynomial of higher degree, which means that these forces can be approximated by adding degrees to the fitted polynomials. We validated experimentally that 3rd and 4th degrees are essential to explain object motion in standard scenarios. Degrees 5 and 6 provide just a small improvement, whereas degrees higher than 6 tend to overfit.

4.1. Splitting into Segments

When tracking fast moving objects in long-term scenarios, objects commonly move back and forth, especially in rallies. During their motion, FMOs abruptly change direction due to contact with players or when they bounce off static rigid bodies. The first step is splitting the sequence into differentiable parts, i.e. detecting *bounces* – abrupt changes of object motion due to contact with other stationary or moving objects. Parts of the sequence between bounces are called *segments*. Segments do not contain abrupt changes of motion and can be approximated by polynomial functions. Theoretically, causal TbD could detect bounces by fitting piecewise linear functions in one frame, but usually the blur is noisy and detecting bounces in just one frame is unstable. This inherent TbD instability can be fixed by non-causal processing.

Figure 4.1. Processing steps of the non-causal Tracking by Deblatting. *Top row*: the causal TbD output with trajectories for all frames overlaid on the first frame. Trajectory-IoU accuracy measure is colour coded from red (failure) to green (success) by scale (Figure 4.3). *Middle rows*: splitting TbD output into segments and fitting polynomials to segments. *Bottom row*: final TbD-NC output. Colour coding: bounces between segments (magenta), bounces between non-intersecting parts (red), fitted polynomials (green), extrapolation to the first and second frame (yellow). Arrows indicate motion direction. Best viewed when zoomed in a reader.

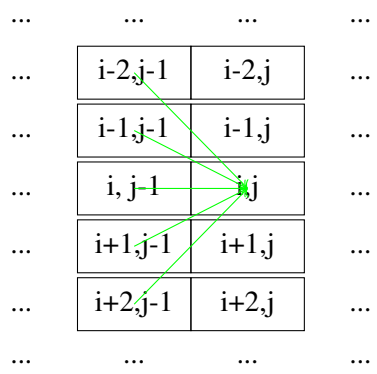


Figure 4.2. Example of dynamic programming. Columns are processed from left to right and 5 neighbouring rows in previous column are used as candidates for trajectory estimate.

To find segments and bounces, we split the whole sequence into *non-intersecting parts* where the object does not intersect its own trajectory, i.e. either horizontal or vertical component of motion direction has the same polarity. Between non-intersecting parts we always report bounces. Bounces inside non-intersecting parts are found by dynamic programming which is able to detect abrupt changes of motion and segments.

The segment between bounces forms an interval between frame t_{s-1} and t_s. Dynamic programming is used to convert blur kernels H_t from all frames in the given non-intersecting part into 1D signal of continuous points. Our aim is to create an object trajectory function $\mathcal{C}_f(t)$, which is continuous in the whole sequence and non-differentiable only at bounces. The proposed dynamic programming approach finds the global minimum of the following energy function

$$E(P) = -\sum_{x=x_b}^{x_e}\sum_{t=t_{s-1}}^{t_s} H_t(x, P_x) + \kappa_1 \sum_{x=x_b+2}^{x_e} \Big|(P_x - P_{x-1}) - (P_{x-1} - P_{x-2})\Big| \\ +\kappa_2(\mathcal{C}_{t_{s-1}}(0) - P_{x_b}) + \kappa_3(P_{x_e} - \mathcal{C}_{t_s}(1))\,, \tag{4.2}$$

where variable P is a discrete 1D version of trajectory $\mathcal{C}$ and it is a mapping which assigns y coordinate to each corresponding x coordinate. The first term is a data term of estimated blur kernels in all frames with the negative sign in front of the sum which accumulates more values from blur kernels while our energy function is being minimised. The second term penalises direction changes and it is defined as the difference between directions of two following points and it is an approximation of the second derivative. The difference is defined as a change in y coordinate and only directions -2, -1, 0, +1, +2 are considered as shown in Figure 4.2. This term makes trajectories smoother and κ_1 serves as a smoothing parameter, which was experimentally set to 0.1. The last two terms enforce that the starting point and the ending point are not far from the ones in the non-intersecting part. Note that the sign in the last two terms is different, because they try to make trajectories shorter and they compete with the first term which prefers longer trajectories, e.g. either making trajectory longer is worth it in terms of values in blur kernels. Parameters κ_2 and κ_3 were both set to 0.1.

Discrete trajectory P is defined from x_b until x_e and these two variables are also being estimated. They are implemented by additional fictional rows, i.e. in each step every point is tested on being a starting or ending point.

The energy function in (4.2) is minimised by a dynamic programming (DP) approach, where accumulated blur kernels H_t are sorted column-wise (H_t) or row-wise (H_t transpose) to account for camera rotation or objects travelling from top to bottom. For both options we find

Figure 4.3. Trajectory recovery for all sequences from the TbD dataset. Trajectory Intersection over Union (TIoU (5.1)) with ground truth trajectories from a high-speed camera is colour coded by the scale on the left. Arrows indicate the direction of motion.

the global minimum of (4.2) and the one with lower energy is chosen. Let us illustrate the DP approach for the column-wise sorting. The row-wise case is analogous. DP starts with the first column and for each pixel in the second column, the best pixel in the previous column is found, which minimises the energy. Consequently, we store the best previous pixel for each row in each column. When all columns are checked, the best trajectory is estimated by backtracking. First, we find a point which gives the lowest energy, which is not necessary in the last column as we check for ending point in every step. Then backtracking is done until the minimising next pixel is in the fictional "starting" row.

When each non-intersecting part is converted into 1D signal, it becomes easier to find bounces. We are looking for points with abrupt changes of direction. When w pixels to the left and w pixels to the right of the given point have a change of direction higher than some threshold, then this point is considered a bounce. After this step, the sequence is split into segments which are separated by bounces.

4.2. Fitting Polynomials

The output discrete trajectory P has a two-fold purpose. It is used first for estimating bounces and segments, and second for estimating which frames belong to the segment and should be considered for fitting polynomials. To this end, we assign starting and ending points of each

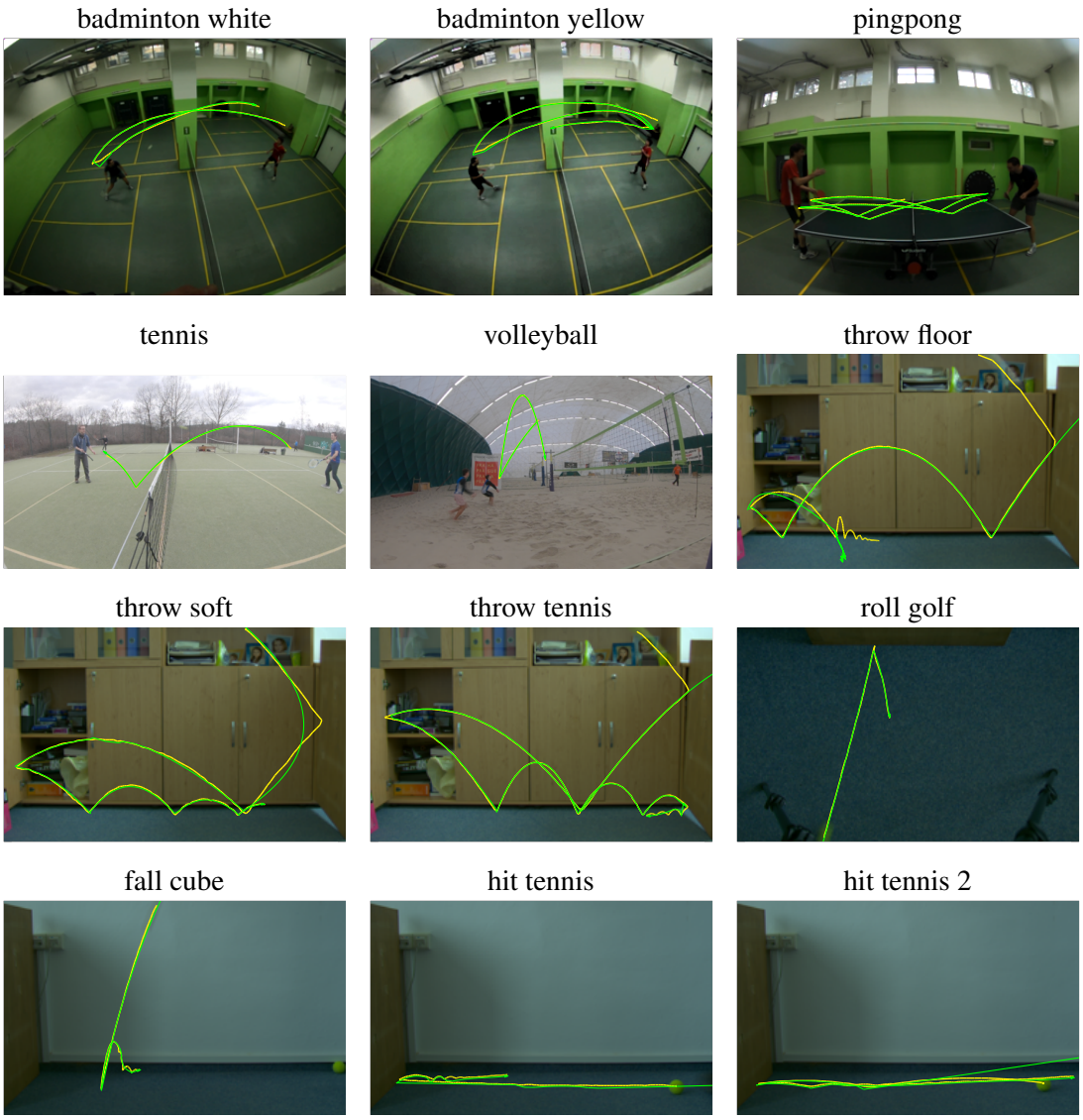


Figure 4.4. Trajectory recovery by the non-causal TbD (TbD-NC) for all sequences from the TbD dataset. Estimated trajectories are shown in green colour. The ground truth trajectory from a high-speed camera is shown in yellow, mostly under the estimated trajectory. Trajectories estimated by TbD-NC are calculated from the causal TbD output (Figure 4.3). Arrows indicate the direction of motion. Names of sequences are shown above each image.

frame, $\mathcal{C}_t(0)$ and $\mathcal{C}_t(1)$, to the closest segment. For fitting, we use only frames that completely belong to the segment, i.e. $\mathcal{C}_t(0)$ and $\mathcal{C}_t(1)$ are closer to this segment than to any other. The degree of a polynomial is a function of the number of frames (N_s) belonging to the segment

$$d_s = \min(6, \lceil N_s/3 \rceil). \tag{4.3}$$

The polynomial coefficients are found by solving a linear least-squares problem

$$\begin{aligned} \min \quad & \textstyle\sum_{t=t_{g-1}}^{t_g} \int_0^1 \|\mathcal{C}_f(t + t_0\epsilon) - \mathcal{C}_t(t_0)\| \mathrm{d}t_0 \\ \text{s. t.} \quad & \mathcal{C}_f(t_{g-1}) = \mathcal{C}_{t_{g-1}}(0) \\ & \mathcal{C}_f(t_g + \epsilon) = \mathcal{C}_{t_g}(1), \end{aligned} \tag{4.4}$$

and after discretising the time into only 2 points (start and end point), we have

$$\begin{aligned} \min \quad & \textstyle\sum_{t=t_{s-1}}^{t_s} \|\mathcal{C}_f(t) - \mathcal{C}_t(0)\|^2 + \|\mathcal{C}_f(t + \epsilon) - \mathcal{C}_t(1)\|^2 \\ \text{s. t.} \quad & \mathcal{C}_f(t_{s-1}) = \mathcal{C}_{t_{s-1}}(0) \\ & \mathcal{C}_f(t_s + \epsilon) = \mathcal{C}_{t_s}(1), \end{aligned} \tag{4.5}$$

where s denotes the segment index. Equality constraints force continuity of the curve throughout the whole sequence, i.e. we get curves of differentiability class C^0. The least-squares objective enforces similarity to the trajectories estimated during the causal TbD pipeline. The final trajectory $\mathcal{C}_f$ is defined over the whole sequence and the last visible point in the frame t which is $\mathcal{C}_t(1)$ corresponds to $\mathcal{C}_f(t+\epsilon)$ in the sequence time-frame, where the exposure fraction ϵ is assumed to be constant in the sequence. The exposure fraction is estimated as an average ratio of the length of trajectories $\mathcal{C}_t$ in each frame and the distance between adjacent starting points

$$\epsilon = \frac{1}{N-1} \sum_{t=1}^{N-1} \frac{\|\mathcal{C}_t(1) - \mathcal{C}_t(0)\|}{\|\mathcal{C}_{t+1}(0) - \mathcal{C}_t(0)\|}. \tag{4.6}$$

Frames which are only partially in segments contain bounces. We replace them with a piecewise linear polynomial which connects the last point from the previous segment, bounce point found by dynamic programming and the first point from the following segment. Frames between non-intersecting parts are also interpolated by piecewise linear polynomial which connects the last point of the previous segment, point of intersection of these two segments and the first point of the following segment. Frames which are before the first detection or after the last non-empty $\mathcal{C}_t$ are extrapolated by the closest segment. Figure 4.1 shows an example of splitting a sequence into segments which are used for fitting polynomials. More examples of full trajectory estimation are in Figures 4.3 and 4.4.

CHAPTER 5
EXPERIMENTS

All versions of Tracking by Deblatting mentioned in Chapters 3 and 4 are evaluated on a newly created TbD dataset. The proposed TbD dataset contains ground truth trajectories from a high-speed camera. Comparing all version of TbD serves as an ablation study. The best-performing version of TbD is compared to the state-of-the-art methods both in classical visual object tracking and fast moving object tracking. The same comparison to state-of-the-art is performed on the FMO dataset [RKŠ+17] which is the first dataset of fast moving objects. Unfortunately, ground truth trajectories for this dataset are not available and the accuracy of trajectory estimation cannot be properly measured. We report only precision and recall of successful detections with non-zero overlap with the ground truth masks. The TbD dataset also contains frames with slow and still objects and this extended TbD (eTbD) version of TbD dataset is used for testing all-speed performance.

We show the results of Tracking by Deblatting and compare it with other trackers on the task of long-term tracking of motion-blurred objects in real-life video sequences. As a baseline, we chose the FMO detector (FMOd [RKŠ+17]), specifically proposed for detecting fast moving objects, and the Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF [LVC+17]) tracker which performs well on standard benchmarks such as VOT [K+19]. CSR-DCF was not designed to track objects undergoing large changes in velocity within a single frame and would perform poorly in the comparison. We therefore augmented CSR-DCF by FMOd reinitialisation every time it outputs the same bounding box in consecutive frames, which is considered a fail. We use FMOd for automatic initialisation to avoid manual input and we skip the first two frames of every sequence to establish background B and initialise CSR-DCF. The rest of the sequence is processed causally (except of TbD-NC), B is estimated as a moving median of the past 3 – 5 frames. To achieve long-term property, we also compare to FuCoLoT tracker [LČZV+18] which is a long-term extension of CSR-DCF tracker.

5.1. TbD Dataset

The comparison with the baseline methods was conducted on a new dataset consisting of 12 sequences with different objects in motion and setting. The settings include different kinds of sports, objects in flight or objects rolled on the ground. Sequences were acquired in both indoor and outdoor scenarios. The sequences contain abrupt changes of motion, such as bounces and interactions with players, and a wide range of speeds. There are sequences where objects are thrown, rolled, hit or just falling or used for playing a particular sport. Sports include badminton, pingpong, tennis, floorball and volleyball. All sequences are listed in Table 5.1

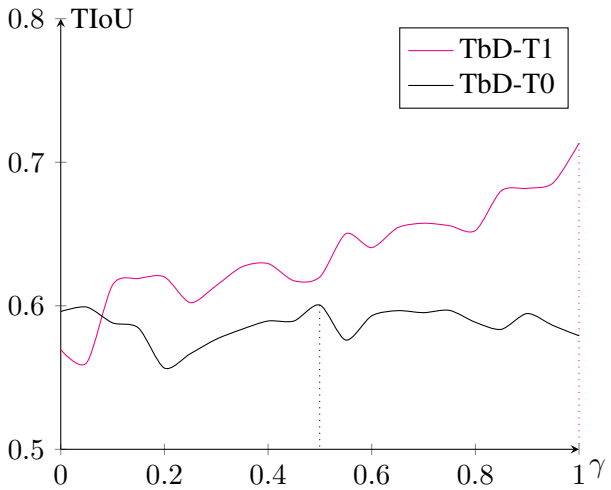


Figure 5.1. Exponential forgetting factor estimation for TbD-T0 and TbD-T1 methods. The graph compares performance in terms of Trajectory-IoU over a subset of the TbD dataset with varying exponential forgetting factors for updating the object model. TbD-T0 has no object template and the best performance is achieved for $\gamma = 0.5$. TbD-T1 was provided with the object template and the best performing setting was for $\gamma = 1$.

with the number of frames in each of them. The extended version (eTbD) is listed in Table 5.5 with the higher number of frames.

The dataset is annotated with the ground-truth trajectory obtained from a high-speed camera footage at 240 frames per second. In comparison, used sequences in the TbD dataset have 30 frames per second. The ground truth for each frame in the standard footage consists of 8 points, sampling the ground truth trajectory with 8 times finer precision. The object in the first frame of the high-speed shooting is marked manually by a bounding box. Then a standard tracker CSR-DCF is used to find the object location in every following frame. Centres of bounding boxes denote ground truth points on the trajectory. In the high-speed shooting, objects do not travel with such a high velocity to be fast moving objects in our definition, because they do not move over distances higher than their size during exposure time. It also means that they are almost not blurred in the high-speed footage. In such scenarios, standard trackers can successfully track the object to create ground truth locations.

We compare the method performance in estimating the motion trajectory in each frame. We therefore generalise Intersection over Union (IoU), the standard measure of position accuracy, to trajectories and define a new measure *Trajectory-IoU* (TIoU):

$$\mathrm{TIoU}(\mathcal{C}, \mathcal{C}^*; M^*) = \int_t \mathrm{IoU}\left(M^*_{\mathcal{C}(t)},\, M^*_{\mathcal{C}^*(t)}\right) \mathrm{d}t, \tag{5.1}$$

where $\mathcal{C}$ is the estimated trajectory, $\mathcal{C}^*$ is the ground-truth trajectory, M^* is a mask with true object appearance obtained from the ground truth, and M_x denotes M placed at location x. TIoU can be regarded as the standard IoU averaged over each position on the estimated trajectory. In practice, we discretise the exposure time into evenly spaced timestamps and calculate intersection over union of the ground-truth object location and the output of a tracker at each time stamp. Then these measurements are averaged. Because the ground truth from the high-speed camera footage was acquired at 8 times higher frame rate, we split exposure time into 8 parts. FuCoLoT and CSR-DCF trackers only output positions, so in this case we estimate linear trajectories from positions in neighbouring frames and then calculate TIoU. FMO detector outputs only non-oriented linear trajectories in each frame independently. In order to calculate TIoU, we need the curve orientation. To this end, we try both orientations (+1 and -1) and

Sequence	TbD-T0, 0		TbD-T0, 0.5		TbD-T1, 1		TbD-NC		TbD-O
	TIoU	Rcl	TIoU	Rcl	TIoU	Rcl	TIoU	Rcl	TIoU
badminton_white	.659	0.92	.657	0.92	.694	0.97	.783	1.00	.792
badminton_yellow	.615	0.89	.626	0.89	.677	0.91	.780	1.00	.788
pingpong	.581	0.89	.590	0.89	.523	0.91	.643	1.00	.697
tennis	.596	0.92	.554	0.89	.673	0.97	.750	1.00	.827
volleyball	.552	0.87	.591	0.90	.795	0.97	.857	1.00	.836
throw_floor	.760	1.00	.776	1.00	.810	1.00	.855	1.00	.864
throw_soft	.584	0.90	.564	0.90	.652	0.97	.761	1.00	.707
throw_tennis	.693	1.00	.777	1.00	.850	1.00	.878	1.00	.872
roll_golf	.414	1.00	.346	1.00	.873	1.00	.894	1.00	.898
fall_cube	.597	1.00	.590	1.00	.721	1.00	.757	1.00	.744
hit_tennis	.564	0.93	.570	0.93	.667	0.93	.714	1.00	.828
hit_tennis2	.476	0.83	.496	0.83	.616	0.83	.682	0.92	.738
Average	.591	0.93	.595	0.93	.713	0.96	.779	0.99	.799

Table 5.1. Ablation study on the TbD dataset. Trajectory Intersection over Union (TIoU) and Recall (Rcl) – comparison of different TbD versions: TbD without template and with exponential forgetting factors (3.2) $\gamma = 0$ (TbD-T0, 0) and $\gamma = 0.5$ (TbD-T0, 0.5), TbD with template and $\gamma = 1$ (TbD-T1, 1), non-causal TbD-T1,1 (TbD-NC) and TbD with oracle (TbD-O). TbD-O shows the highest attainable TIoU for the TbD core as a reference point when predictions are precise. The highest TIoU for each sequence is highlighted in blue colour and the highest recall in cyan colour. When TbD-NC outperforms TbD-O, the score is highlighted in red.

report the highest TIoU. For standard trackers orientation is given from the centres of bounding boxes. Estimating orientation is part of the proposed TbD method. In the beginning, when just a detection by FMOd is given, the orientation is not known. After the prediction is done in both directions, the orientation with the highest fitting score of TbD is chosen.

We evaluated four flavours of TbD that differ in the presence of the initial user-supplied template $\hat{F}$, the learning rate γ of the object model in (3.2) and non-causal trajectory estimation. The presented flavours are:

- TbD-T0,0: Template not available, model update is instantaneous (memory-less), $\gamma = 0$.
- TbD-T0,0.5: Template not available, model is updated with the learning rate $\gamma = 0.5$.
- TbD-T1,1: Template available, model remains constant and equal to the template, $\gamma = 1$.
- TbD-NC: non-causal TbD-T1,1 with full trajectory estimation (Chapter 4).

Empirical justification of chosen learning rates is presented in Figure 5.1. We evaluated all learning rates from 0 to 1 with the step size 0.05 for each method, i.e. TbD-T1 and TbD-T0. For each step size, the average TIoU was computed over a subset of the TbD dataset and the best performing setting was chosen. When template is not available, updating model smoothly with the rate $\gamma = 0.5$ dominates instantaneous update ($\gamma = 0$) and no update at all ($\gamma = 1$), i.e. keeping the first estimate. When template is available, it is preferable to keep the template rather than updating it. Even when no update is done ($\gamma = 1$), it is still preferable to minimise the loss (3.3) with respect to F. Template $\hat{F}$ usually contains only object-specific details. However, image noise or other phenomena as shadows should be explained by some variables and minimisation with respect to F can serve this purpose. For instance, we would like to include shadows or prompt illumination changes into the object model F, but updating the template and learning such noise is not desirable. Figure 5.1 (TbD-T0) has two local maxima,

	TIoU		%	
	TbD	TbD-NC	TbD	TbD-NC
TbD Fails	0.000	0.382	4.7	0.4
TbD TIoU>0	0.744	0.800	95.3	99.6

Table 5.2. Comparison of non-causal TbD (TbD-NC) with causal TbD. TbD failure is defined as frames where Trajectory-IoU equals to zero. TbD-NC decreases the number of frames with failure by a factor of 10.

one at 0.5 and one at a point near zero. It means that TbD-T0,0 and TbD-T0,0.5 are two versions of TbD with no template which give reasonable performance.

Comparison of all versions of the TbD method is shown in Table 5.1 in form of an ablation study, sorted from left to right by its performance. Performance is measured by a newly proposed Trajectory-IoU score and the traditional recall measure, which is a ratio of correctly found true positive detections over all positives / detections. Detection is called a true positive if it has non-zero overlap with the ground truth. To evaluate the performance of the core part of TbD that consists of deblatting and trajectory fitting alone, we provide results of a special version of the proposed method called "TbD with oracle" (TbD-O). This behaves like regular TbD but with a perfect trajectory prediction step. We use the ground-truth trajectory to supply the region D to the deblatting step exactly as if it were predicted by the prediction step, effectively bypassing the long-term tracking logic of TbD. The rest is identical to TbD-T1,1. TbD with oracle tests the performance and potential of the deblatting and trajectory estimation alone because failures do not cause long-term damage – success in one frame is independent of success in the previous frame. It serves as a reference point of what deblatting and fitting in one frame can achieve if everything else is given. However, TbD-O can not be used in real-life scenarios and we use the best-performing TbD-NC in further experiments.

Table 5.2 shows that TbD-NC corrects complete failures of causal TbD when TIoU is zero, e.g. due to wrong predictions or other moving objects. TbD-NC also improves TIoU of successful detection by fixing small local errors, e.g. when the blur is misleading or fitting in one frame is not precise.

Among other TbD flavours, it is no surprise that availability of the object template is beneficial and outperforms other versions. However, even if the template is not available, TbD can learn the object model and updating the appearance model gradually during tracking is preferable to instantaneous updates. Trajectory estimation in sequence (TbD-NC) gives even more boost in performance. This is the only version which can potentially outperform TbD with oracle and estimate trajectories more accurately by non-causal post-processing of all frames jointly. This indeed happens in four cases in Table 5.1, where TbD-NC gives better results than TbD-O. The proposed non-causal TbD outperforms all other TbD flavours in both recall and TIoU. Recall is 100% in all cases except one, where the first detection appeared only on the seventh frame and extrapolation to the first six frames was not successful. TbD-O has 100% recall in all situations by construction. The average TIoU and recall for TbD-O is just marginally higher than for TbD-NC and the gap is small considering that TbD-O knowns exactly where the detection should be.

Table 5.3 presents results of the comparison to the baselines. All versions of Tracking by Deblatting outperform baseline methods on average by a wide margin, both in the traditional recall measure as well as in trajectory accuracy TIoU score. FMO detector is less accurate and more prone to false positives as it lacks any prediction step and by design ignores slow objects. CSR-DCF, despite reinitialisations by FMOd, fails to detect fast moving objects accurately. FuCoLoT is even less accurate, but has higher recall thanks to the long-term property.

Sequence	#	CSR-DCF [LVC+17]		FuCoLoT [LČZV+18]		FMO [RKŠ+17]		TbD-NC (this work)	
		TIoU	Rcl	TIoU	Rcl	TIoU	Rcl	TIoU	Rcl
badminton_white	40	.286	0.39	.286	0.39	.242	0.34	.783	1.00
badminton_yellow	57	.123	0.22	.123	0.22	.236	0.31	.780	1.00
pingpong	58	.064	0.12	.065	0.14	.064	0.12	.643	1.00
tennis	38	.278	0.64	.294	0.89	.596	0.78	.750	1.00
volleyball	41	.533	0.82	.496	0.79	.537	0.72	.857	1.00
throw_floor	40	.287	0.71	.275	0.63	.272	0.37	.855	1.00
throw_soft	60	.470	0.97	.463	0.95	.377	0.57	.761	1.00
throw_tennis	45	.444	0.95	.239	0.98	.507	0.65	.878	1.00
roll_golf	16	.331	1.00	.360	1.00	.187	0.71	.894	1.00
fall_cube	20	.324	0.67	.324	0.67	.408	0.78	.757	1.00
hit_tennis	30	.329	0.93	.330	0.93	.381	0.68	.714	1.00
hit_tennis2	26	.214	0.79	.226	0.79	.414	0.71	.682	0.92
Average	39	.307	0.68	.290	0.70	.352	0.56	.779	0.99

Table 5.3. Trajectory Intersection over Union (TIoU) and Recall (Rcl) on the TbD dataset – comparison of the best performing TbD method (TbD-NC, see Table 5.1) to the state-of-the-art methods: CSR-DCF [LVC+17] tracker, FuCoLoT [LČZV+18] tracker and the Fast Moving Object method [RKŠ+17]. CSR-DCF is a standard, well-performing [K+19], near-real time tracker. FuCoLoT is a long-term extension of CSR-DCF. For each sequence, the highest TIoU (5.1) is highlighted in blue and recall in cyan. The number of frames is indicated by "#" sign.

A visual demonstration of tracking by the proposed method is shown in Figure 4.3 (for TbD-T1) and in Figure 4.4 (for TbD-NC). Trajectory-IoU for the causal TbD is visualised in colour ranging from red (TIoU = 0) through yellow (TIoU = 0.3) up to green (TIoU = 1). Trajectory estimation in sequence (TbD-NC) reconstructs more precise trajectories than without it (TbD-T1). Only in frames where the object slows down, dynamic programming approach is not robust and non-causal trajectory estimation could fail. Such situation and other failures can be detected by checking the average error of non-causal fitting. For instance, the throw floor sequence in Figure 4.4 contains segments in the end of the sequence when the object is slow and bounces a lot. In such case, there is a big deviation between TbD-NC and TbD outputs as in Figure 4.3 and the segment is not replaced by non-causal fit, e.g. the output of causal TbD is used for evaluation. The first segment in the beginning was not able to extrapolate to the first two frames successfully, as the bounce in the second frame was not detected. Similarly, the throw soft sequence in Figure 4.4 shows a failure of dynamic programming where a bounce was not successfully detected and thus just the output of causal TbD from Figure 4.3 was used.

5.2. FMO Dataset

FMO dataset [RKŠ+17] was introduced as the first dataset containing only fast moving objects, now at version 2. The FMO dataset does not contain ground-truth trajectory data, but only binary masks which denote regions affected by fast moving objects, annotated by hand. Therefore, the trajectory accuracy cannot be evaluated and we report traditional precision/recall measure, which is derived from the intersection of detection and the ground-truth mask. Detection is considered successful if it has non-zero overlap with the ground truth mask. On the FMO dataset, the TbD method is slightly better than the FMO method in recall, owing to the fact that the initial detection is done by FMOd and if FMOd fails then TbD cannot start track-

Sequence name	#	FMO [RKŠ+17]		TbD-T0, 0.5	
		Precision	Recall	Precision	Recall
volleyball1	50	100.0	45.5	100.0	70.0
volleyball passing	66	21.8	10.4	72.7	48.5
darts	75	100.0	26.5	100.0	0.0
darts window	50	25.0	50.0	100.0	0.0
softball	96	66.7	15.4	53.9	25.0
archery	119	0.0	0.0	0.0	0.0
tennis serve side	68	100.0	58.8	93.3	77.8
tennis serve back	156	28.6	5.9	100.0	44.0
tennis court	128	0.0	0.0	100.0	0.0
hockey	350	100.0	16.1	0.6	1.6
squash	250	0.0	0.0	100.0	0.0
frisbee	100	100.0	100.0	100.0	100.0
blue ball	53	100.0	52.4	100.0	66.7
ping pong tampere	120	100.0	88.7	95.8	88.2
ping pong side	445	12.1	7.3	95.1	55.7
ping pong top	350	92.6	87.8	90.2	79.6
Average	154	59.2	35.5	81.6	41.1

Table 5.4. Precision and recall on the FMO dataset of the TbD tracker (setting: TbD without template and with exponential forgetting factor $\gamma = 0.5$) and the FMO method [RKŠ+17], average on the 16 sequences of the FMO dataset.

ing, but significantly better in terms of precision. Table 5.4 shows aggregated results on all 16 sequences. The number of frames is indicated by "#" sign. Sequences in the FMO dataset are much larger and the evaluation took around 20 hours, compared to 1 hour on the TbD dataset. The TbD dataset contains only the most interesting parts of sequences and unnecessary frames are cropped, but added to the extended TbD dataset.

The main drawback of the FMO dataset is its lack of ground truth trajectories. Even completely wrong trajectories, when convolved with the object mask, can lead to perfect overlap with the ground truth. For instance, if the output trajectory is identical to the real trajectory in the first part, but then it comes back to the starting point, thus the estimated trajectory is twice longer. The output trajectory is completely wrong and cannot be used for predicting the object location in the next frame, but it will produce a 100 % overlap with the ground truth object location mask. Another example is a trajectory which is oriented in the opposite direction. Ground truth in the style of the FMO dataset will give 100 % accuracy, whereas ground truth in the new style of exact trajectories as in the TbD dataset will give close to 0 % accuracy if trajectories are long enough. Ground truth masks in the FMO dataset do not contain fine details about the object location, in comparison to the TbD dataset.

There are no object templates provided in the FMO dataset, thus TbD-T0 version was used. For some sequences, where the object is never slow and sharp enough, the real object mask is not even precisely known.

A visual demonstration of tracking by the proposed method on some sequences of the FMO dataset is shown in Figure 5.2. Each image depicts trajectories from all frames, superimposed on a single image from the sequence. Arrows indicate the direction of motion. Standard intersection over union is encoded by colour, from green (IoU=1) to red (IoU=0, false positive). Trajectories are estimated successfully with the exception of frames where the object is in direct contact with other moving objects, which throws off the local estimation of the background.

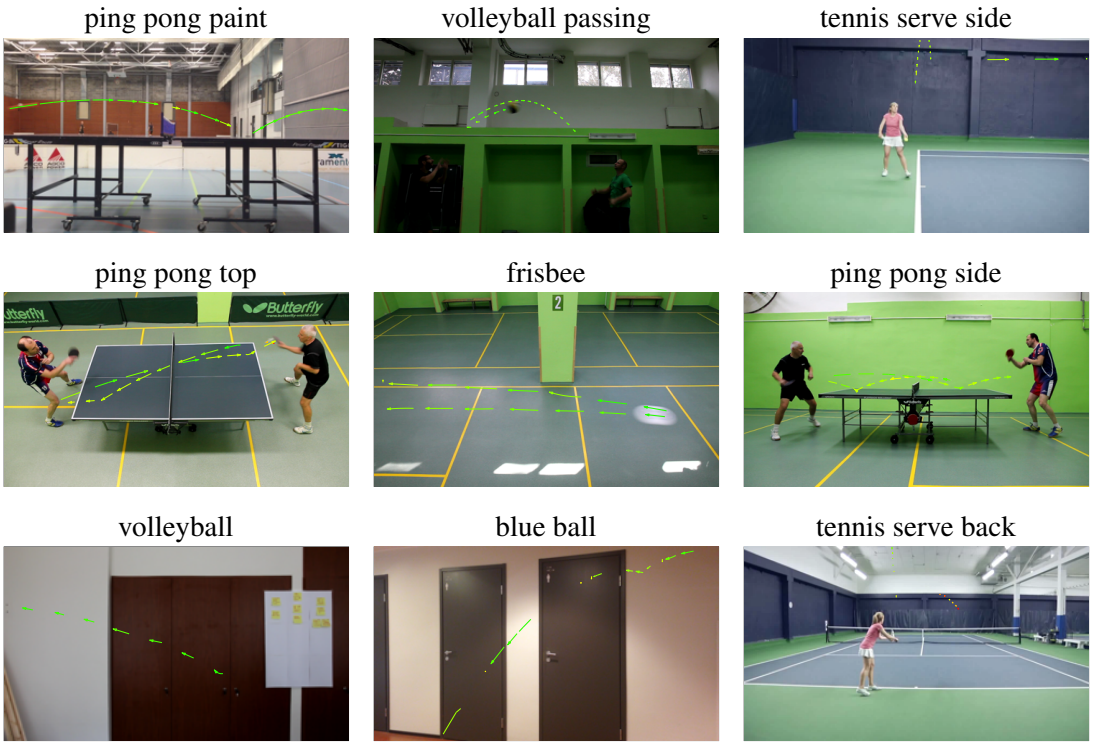


Figure 5.2. Trajectory recovery for 9 selected sequences from the FMO dataset [RKŠ+17]. Intersection over Union (IoU) with the ground truth occupancy mask is colour coded using the scale from Figure 4.3. Arrows indicate the direction of motion. Names of sequences are shown above each image.

We do not show full trajectories estimated by non-causal post-processing as this is not directly possible in the FMO dataset. In many sequences, several different objects of the same class are visible and thus trajectories are not continuous. For instance, players use one ping pong ball and when it falls down, they start playing with another ball. This shows the limitations of non-causal post-processing.

Some sequences in the FMO dataset contain a lot of camera motion. The original FMO method [RKŠ+17] used camera stabilisation to account for that. TbD method in its core computes the background by moving median of several last frames. For fairness, we also added camera stabilisation into the proposed TbD method.

5.3. All-speed Tracking

The inner part of the TbD method consists of deblatting and fitting which allow estimating robust intra-frame object locations. Speed of the object can be arbitrary, albeit performance is better for higher speed when the object is not perfectly round and homogeneous. We evaluated the performance of the TbD-NC method on the extended TbD dataset (eTbD), which contains the same sequences as the TbD dataset but with on average around twice more frames with objects slowing down and staying still. Originally, the eTbD dataset was created first and the TbD dataset was made by cropping the eTbD dataset, such that all speeds are represented equally.

For normalisation, we represent speed in radii per exposure which measures the number of radii the object travels in one exposure time. Speeds less than one radii per exposure $[r/\epsilon]$,

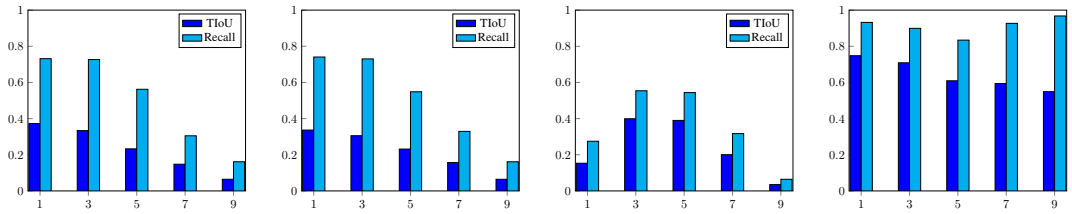


Figure 5.3. All-speed tracking. Trajectory-IoU and recall on the extended TbD dataset (eTbD) for different algorithms (from left to right) – CSR-DCF [LVC+17], FuCoLoT [LČZV+18], FMO algorithm [RKŠ+17] and non-causal Tracking by Deblurring (TbD-NC). The horizontal axis denotes speed which is measured in radii per exposure. The vertical axis shows both success rates measured by Trajectory-IoU (5.1) and recall.

Figure 5.4. Objects with varying speeds (0, 1, 3, 5, 7, 9) in radii per exposure, which removes dependence on camera settings and object size.

i.e. not FMOs, represent half of frames in the eTbD dataset and the other half contains FMOs. Table 5.5 shows results on the eTbD dataset for the TbD-NC method and compares it to other baselines.

In Figure 5.3, we report histograms of performance of all-speed tracking for every method, measured by the average TIoU in blue and by recall in cyan. Histogram bins represent different speeds ranging from 1 to 9 radii per exposure. Standard trackers such as CSR-DCF and FuCoLoT have similar performance which declines quickly for higher speeds. The FMO method [RKŠ+17] has peek performance for speeds between 3 and 5 radii per exposure, lower or higher speeds decrease TIoU and recall drastically. FMO method is based on difference images and very high speeds cause low contrast images and the object becomes almost invisible in the difference image. On the other side, the FMO method was not designed to track not so fast moving objects and its performance drops for slow objects. The TbD method solves both problems and indeed connects the world of fast moving objects and the world of slow or still objects. For very high speeds, the TbD method does not suffer from low contrast images because the image formation model is still valid. TbD-NC has a bit decreasing TIoU for higher speeds, but its recall is close to one in all cases. Lower TIoU for higher speeds can be explained by the difficulty of deblatting and fitting when the object is severely blurred. When a severely blurred object has a colour similar to the background, the part of the loss function which minimises L^1 norm of the blur kernel will try to avoid explaining the motion caused by the object. For sequences where this is the case, we lowered the weights of the total variation term which enforces sparsity of the blur kernel and reduces small non-zero values.

All-speed tracking posed another problem of estimating background when the object is close

Sequence (extended)	#	FuCoLoT [LČZV+18]		FMO [RKŠ+17]		TbD-NC (this work)	
		TIoU	Recall	TIoU	Recall	TIoU	Recall
badminton_white	125	.232	0.40	.142	0.19	.635	0.85
badminton_yellow	125	.155	0.33	.229	0.30	.536	0.84
pingpong	95	.062	0.10	.100	0.15	.604	0.98
tennis	118	.245	0.84	.554	0.74	.420	0.58
volleyball	72	.500	0.79	.430	0.56	.814	0.97
throw_floor	73	.147	0.34	.153	0.21	.896	1.00
throw_soft	150	.516	0.98	.303	0.51	.790	1.00
throw_tennis	71	.232	0.99	.347	0.46	.867	1.00
roll_golf	16	.360	1.00	.187	0.71	.894	1.00
fall_cube	28	.414	0.77	.341	0.65	.759	1.00
hit_tennis	57	.330	0.96	.225	0.42	.772	1.00
hit_tennis2	26	.226	0.79	.414	0.71	.681	0.92
Average	80	.285	0.69	.285	0.47	.722	0.93

Table 5.5. Trajectory Intersection over Union (TIoU) and Recall (Rcl) on the eTbD dataset. Extended version of the TbD dataset is used to evaluate the performance of TbD-NC in long-term scenarios and on objects with different speeds, ranging from still objects to very fast moving objects. The number of frames is denoted by "#" sign. The proposed non-causal Tracking by Deblatting (TbD-NC) performs better than the baselines FuCoLoT and the FMO method. TIoU and Recall are lower than on the TbD dataset (Table 5.3) due to more challenging tasks in the eTbD dataset.

to still. The median of previous several frames is not sufficient. To this end, we increased the number of frames used for estimating the background to 20 previous frames, which is used when object speed is less than a threshold. For still objects with zero speed, the background is not updated.

5.4. Speed Estimation

TbD-NC provides the function $\mathcal{C}_f(t)$, which is defined for each real-valued time stamp t between zero and the number of frames. Taking the norm of the derivative of $\mathcal{C}_f(t)$ gives a real-valued function of object velocity, measured in pixels per exposure. To normalise it with respect to the object, we report speed in radii per exposure. This is achieved by dividing the speed by the object radius. Examples of objects with different speed in radii per exposure are presented in Figure 5.4. Intra-frame speed estimation for all sequences from the TbD dataset is visualised in Figure 5.5, where sequences are shown together with their speed functions.

The ground truth speed was estimated from a high-speed camera footage having 8 times higher frame rate. The object centre was detected in every frame and the GT speed was then calculated from the distance between the object centres in adjacent frames. Then, the speed is multiplied by 8 (difference in exposure) and divided by the object radius. Deliberately, we used no prior information (regularisation) to smooth the GT speed and therefore it is noisy as can be seen in Figure 5.5. Two factors influenced this. First, the discrete origin of ground truth which has a noisy derivative. For example, if the object moves with a speed of 2.5 pixels per exposure, the ground truth gives oscillating speeds of 2 and 3 pixels per exposure. This is caused by the output of the standard tracker used for calculating the ground truth from the high-speed camera. Second, the fact that infinitely many speed functions represent the same $\mathcal{C}_f(t)$ function causes some uncertainty in the speed estimation.

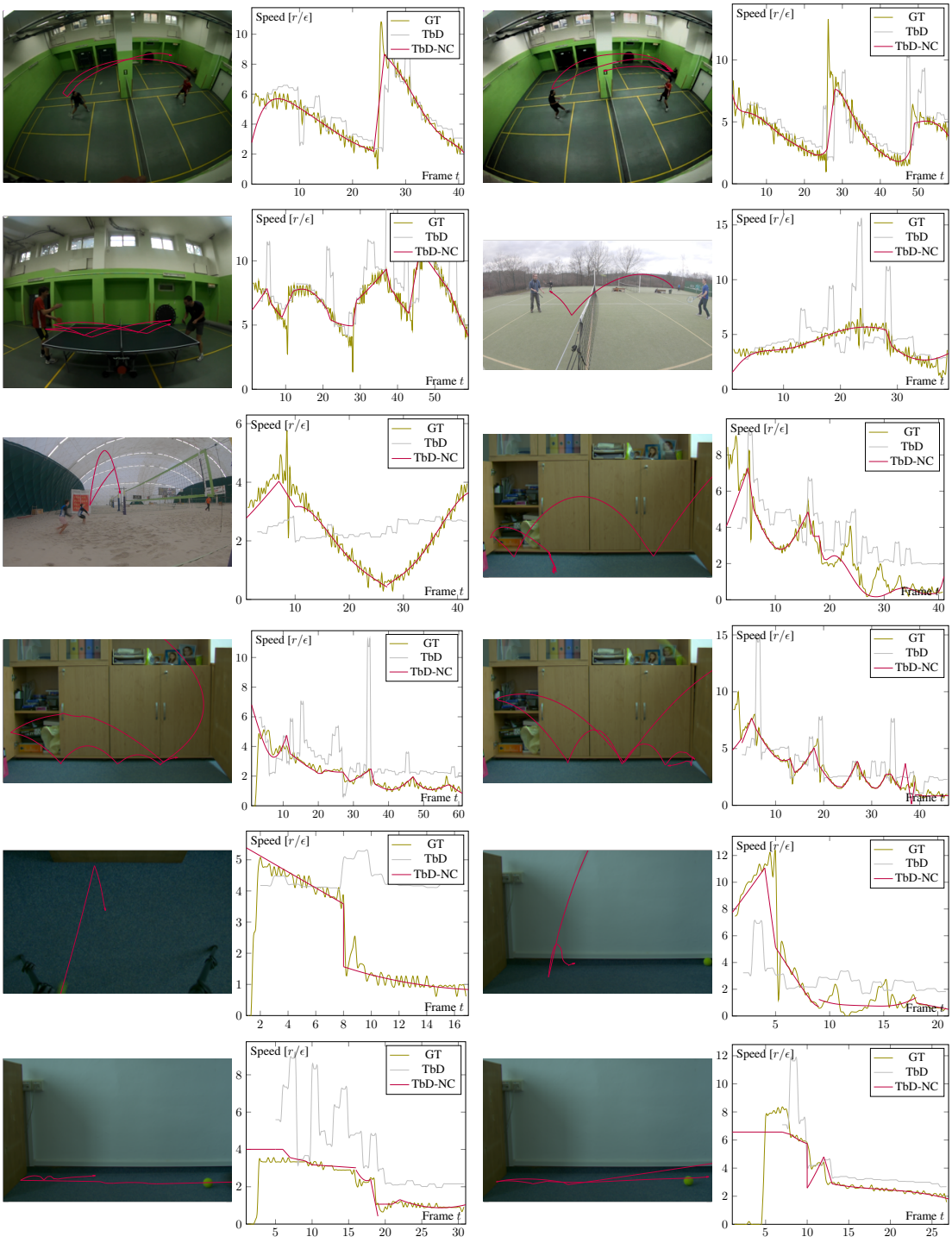
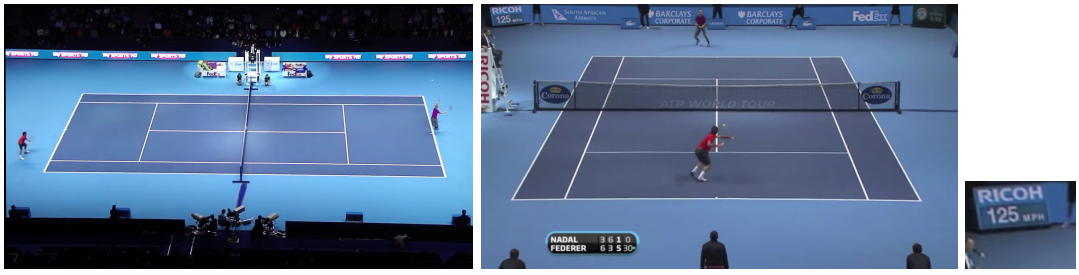


Figure 5.5. Speed estimation using non-causal Tracking by Deblatting (TbD-NC) on all sequences from the TbD dataset. Trajectories estimated by TbD-NC are overlaid on the first frame of each sequence. Graphs contain the speed estimation by the causal TbD method (lightgray) and TbD-NC (purple) in radii per exposure compared to the "ground truth" speed (olive) calculated from a high-speed camera. The noise and oscillations in GT are caused by discretisation. Median differences to GT for all sequences are shown in Table 5.7. The causal TbD has no extrapolation to first frames.

Serve	Duration [frames]	GT [mph]	Hrabalík [Hra17]		TbD-NC	
			Speed [mph]	Error [%]	Speed [mph]	Error [%]
1	23	108	105.6	2.2	108.0	0.0
2	32	101	103.8	2.8	101.6	0.6
3	62	104	106.5	2.4	110.4	6.1
4	75	113	101.7	10.0	115.8	2.5
5	82	104	91.9	11.6	106.9	2.8
6	30	127	127.4	0.3	126.3	0.6
7	34	112	116.1	3.7	107.5	4.0
8	78	125	123.2	1.4	130.3	4.2
9	67	99	88.3	10.8	89.7	9.4
10	90	108	110.2	2.0	106.2	1.6
Mean	57	110.1	107.5	4.7	110.3	3.2

Table 5.6. Speed estimation in a tennis match compared to the radar gun (GT). We used the last 10 serves of the final match of 2010 ATP World Tour. The speed is reported in miles per hour (mph). The lowest error for each serve is marked in blue.



Spectator's view | Another view (speed in top left corner) | Cropped

Figure 5.6. Radar gun measurements. Speed was automatically estimated by the TbD-NC method from the video on the left. Ground truth acquisition from YouTube video is shown in the middle and the right images. Table 5.6 compares estimates to the ground truth.

We also report the median of absolute differences between GT and the estimated speed in Table 5.7. The error is mostly due to the noise in GT. Nevertheless, median error is 0.32 radii per exposure, which is a small error when speeds are in the range of near 10 radii per exposure.

5.4.1. Speed Estimation Compared to Radar Guns

In sports, such as tennis, radar guns are commonly used to estimate the speed of serves. In this case, only the maximum speed is measured and the strongest signal usually happens immediately after the racquet hits the ball.

Hrabalík [Hra17] in his master's thesis gathered the last 10 serves of the final match of 2010 ATP (Association of Tennis Professionals) World Tour. Rafa Nadal and Roger Federer played in this match. The serves were found on YouTube from a spectator's viewpoint[1]. Ground truth was available from another footage which showed the measured speed[2] from radar guns. Hrabalík's version of FMO detector achieved quite precise estimates of the speed, with the

[1] https://youtu.be/3deJOQ0dCDU
[2] https://youtu.be/YCPHpb61Cnk?t=443

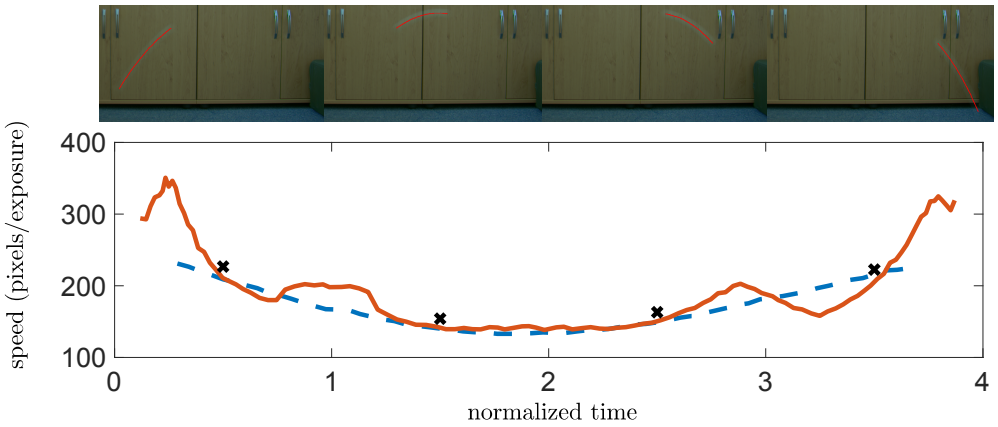


Figure 5.7. Estimating the object velocity from blur kernels. In four consecutive frames (top row), object trajectories were estimated with TbD. The bottom plot shows the velocity during exposure calculated from the blur kernels (solid red) and the ground-truth velocity (dashed blue line) obtained by a high-speed camera. Black crosses show the average velocity per frame calculated from the trajectory length.

average error of 4.7 %, where the error is computed as

$$E_v = \frac{\|v - v_{gt}\|}{v_{gt}}. \tag{5.2}$$

Unfortunately, the ATP footage from spectator's viewpoint is of a very poor quality with tennis ball being visible only as several pixels. Even in his work, Hrabalík used a special set of parameters to make the algorithm work in this setting. Deblurring does not perform well when a video has low resolution or the object of interest is poorly visible. To test only the performance of full trajectory estimation by TbD-NC, we manually annotated the starting and ending points of the ball trajectory in several frames after the hit in every serve. Then we find the time-stamp t_{hit} so that the final trajectory $\mathcal{C}_f(t_{hit})$ at this point is the closest to the hit point. Then $\|\mathcal{C}'_f(t_{hit})\|$ is the speed measured by TbD-NC.

To convert the speed to real world values as reported by radar gun (miles per hour, mph), we used the same approach as in Hrabalík's work. The pixel-to-miles transformation was computed by measuring the court size in the video (1519 pixels) and dividing it by the tennis standards (78 feet). Camera frame rate was used according to the standard of 29.97 fps. Figure 5.6 shows how the ground truth was acquired. Additionally, due to severe camera motion, the video was stabilised.

Table 5.6 compares the speed estimated by TBD-NC and FMO methods to the ground truth from the radar. The proposed TbD-NC method is more precise than the FMO method and in several cases the speed is estimated with GT error close to zero. The estimated average speed over 10 serves differs from the ground truth only by 0.2 mph, which demonstrates that TbD-NC calculates object velocity on par with expensive radar guns.

5.4.2. Speed from Blur Kernel

Apart from estimating speed by taking the norm of the derivative of the fitted function $\mathcal{C}_f(t)$, we can also directly estimate speed from the blur kernel H. The values in the blur kernel are directly proportional to time the object spent in that location. For example, if half of the exposure time the object was moving with a constant velocity and then it stopped and stayed still, the blur kernel will have constant intensity values terminated with a bright spot that will be equal to the sum of intensities of all other pixels. Estimating speed from blur intensity values

Sequence	Speed	Radius			Gravity	
	Median Diff. $[r/\epsilon]$	GT [cm]	Estimate [cm]	Error [%]	Estimate $[ms^{-2}]$	Error [%]
badminton_white	0.41	-	-	-	-	-
badminton_yellow	0.43	-	-	-	-	-
pingpong	0.53	2.00	1.99	0.3	9.53	2.8
tennis	0.39	-	-	-	-	-
volleyball	0.37	10.65	10.47	1.7	10.50	7.2
throw_floor	0.29	3.60	3.47	3.7	10.21	4.2
throw_soft	0.19	3.60	3.72	3.3	9.52	2.9
throw_tennis	0.21	3.43	3.69	7.6	9.19	6.2
roll_golf	0.27	-	-	-	-	-
fall_cube	0.38	2.86	2.63	8.0	10.66	8.8
hit_tennis	0.18	-	-	-	-	-
hit_tennis2	0.24	-	-	-	-	-
Average	0.32	-	-	4.1	9.93	5.3

Table 5.7. Estimation of radius, speed and gravity by the proposed TbD-NC method on the TbD dataset. Trajectories estimated by TbD-NC are used to measure physical properties of the object and the environment. The speed estimates are compared to the ground truth speed from a high-speed camera. Radius is calculated when assuming Earth gravity 9.8 ms^{-2}. Standard object sizes are taken as ground truth for radius. When the radius is known, we compute gravity.

is however not very reliable due to noise in the blur kernel (e.g. camera noise, compression artefacts). Figure 5.7 illustrates a case where this approach works. All pixels in the blur kernel H which lay on the trajectory $\mathcal{C}$ are used for calculating the object velocity.

5.5. Shape and Gravity Estimation

In many situations, gravity is the only force that has non-negligible influence. Then it is sufficient to fit polynomials of second order

$$\mathcal{C}(t) = x_0 + vt + at^2. \tag{5.3}$$

If parameters of the polynomial are estimated correctly, and the real gravity is given, then transforming pixels to metres in the region of motion is feasible. Gravity in the equation (5.3) is represented by a parameter a, which has units $[\text{px}(\frac{1}{f}s)^{-2}]$, where the frame rate is denoted by f. If we assume the gravity of Earth $g \approx 9.8[ms^{-2}]$, f is known and a is estimated by curve fitting, the formula to convert pixels to meters becomes

$$p = \frac{g}{2af^2}, \tag{5.4}$$

where p are meters in one pixel on the object in motion. For example, in our case with approximately round objects, we compute radius in centimetres as $r_{cm} = 100pr$ from estimated radius r in pixels found during deblatting or by FMO detector.

The radius estimation by this approach is shown in Table 5.7. Only half of the TbD dataset is used, i.e. sequences where the object was undergoing only motion given by the gravity (throw, fall, ping pong, volleyball). In other cases such as roll and hit, the gravity has almost no influence and this approach cannot be used. The badminton sequences have large air resistance and the tennis sequence was recorded outside during strong wind. When gravity was indeed

1st frame 3rd frame 6th frame TbD-NC

Figure 5.8. Gravity and shape estimation on a sequence from a web camera. A floor ball was thrown from the top. Three images on the left show individual trajectories estimated by causal TbD. The final trajectory estimated by non-causal TbD is shown on the right, with blue arrows indicating the object location for every integer time t. The final trajectory looks linear, but only its trace is close to linear. The second order term in the fitted polynomial is used to model acceleration given by gravity which allows calculating object shape and the gravity itself.

Dodgeball Beach volleyball Air hockey

Figure 5.9. Examples of three sequences found on YouTube which contain fast moving objects. Estimated object trajectories by TbD from multiple frames are rendered into one frame.

the only strong force, the estimation is quite robust with average error of only 4.1 %. Ground truth was taken from standard sizes of used objects.

Alternatively, when the real object size is known we can instead estimate gravity, e.g. when throwing objects on another planet and trying to guess which planet it is. In this case, (5.4) can be rewritten to estimate g. Results are also shown in Table 5.7 and the average error is 5.3 % when compared to the gravity on Earth. This shows robustness of the approach in both estimating radius and gravity.

The performance of TbD-NC to measure the shape and gravity is tested on an additional sequence from a web camera. A blue floor ball was thrown in front of the camera so that the trajectory is almost linear and perpendicular to the floor. In reality, only the trace of the trajectory is linear, but the fitted polynomial needs to be of a higher order. The second order term in the polynomial is used to model acceleration given by the gravity which allows calculating object shape and the gravity itself by the same approach. Figure 5.8 shows the final estimated trajectory. When we fix Earth gravity 9.8 ms^{-2}, then from the equation (5.4) we get floor ball radius 3.55 cm, which is only 1 % error. When the ground truth radius of 3.6 cm is fixed, then the estimated gravity is 9.93 ms^{-2}, which is again 1 % error. The error is mainly due to the radius estimation in pixels which is computed as half of the size of the estimated object model F.

5.6. Other Applications

Among other applications of the proposed Tracking by Deblatting are fast moving object removal and temporal super-resolution. The task of temporal super-resolution stands for creating a high-speed camera footage out of a standard video and consists of three steps. First, a video free of fast moving objects is produced, which is called fast moving object removal. For all FMOs which are found in every frame, we replace them with the estimated background. Sec-

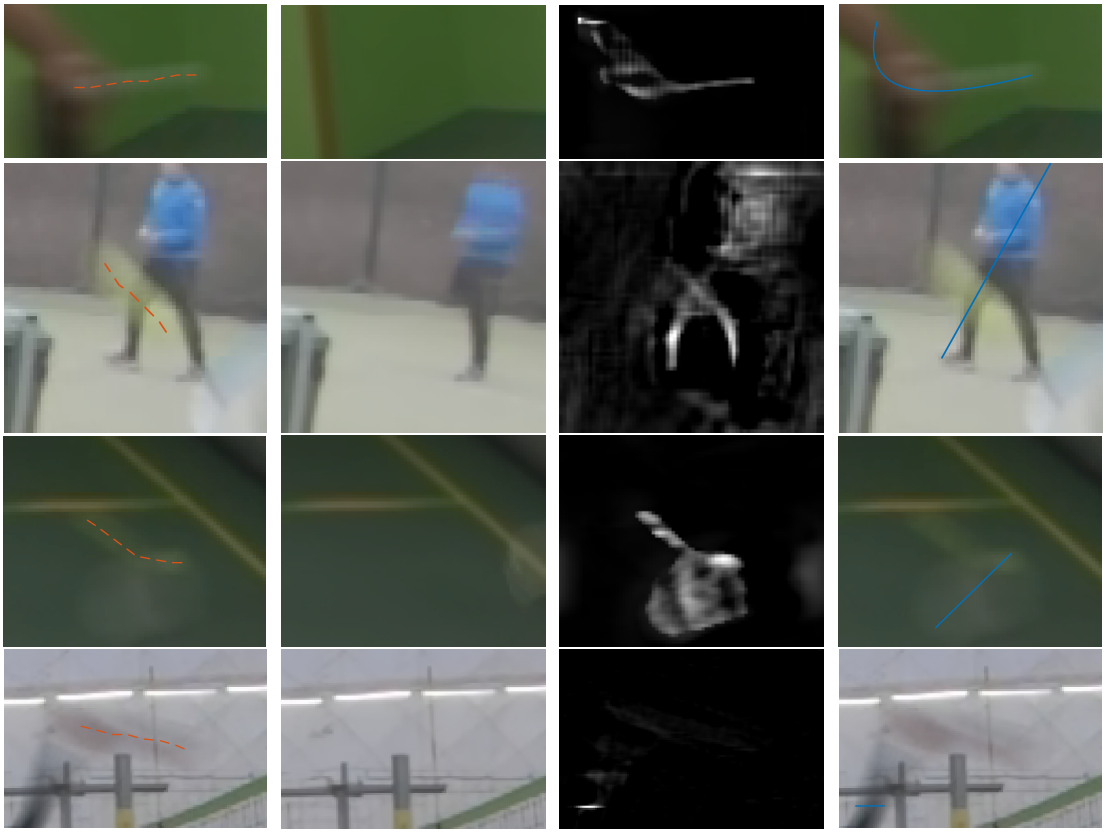


Figure 5.10. Examples of failed trajectory estimation due to interference with other objects (incorrect background B). Left to right: input image with indicated ground-truth trajectory, background estimated by moving median (see the difference to the actual background in the left), estimated PSF with artefacts due to background inconsistency, erroneous trajectory fit. From top to bottom the problems in background (sometime hard to identify) are hand and racket of the player, another player in the far background, badminton racket of the player, and hand in contact with the volleyball.

ond, intermediate frames between adjacent frames are calculated as their linear interpolation. Objects which are not FMOs will look natural after linear interpolation. The FMO trajectory function $\mathcal{C}_f(t)$ is split into the required number of pieces, optionally with shortening to account for the desired exposure fraction. Third, the object model (F, M) is used to synthesise (as in Figure 3.9) the video formation model with FMOs (3.1). Examples of these applications are provided in the supplementary files as videos.

5.7. Limitations

Tracking by Deblatting is still limited by several factors. Mainly due to the complexity of blind deblurring, the method is currently limited to objects that do not significantly change their perceived shape and appearance within a single frame. TbD works best for approximately round and uniform objects. Extension to more complicated shapes as well as greater robustness to interference with other objects is the future work.

Fitting in one frame is not robust to failures when there is other motion in the neighbourhood of the moving object. Other motions create additional points in the blur kernel which should be explained by fitting. When motion caused by the object of interest is dominant, RANSAC used in fitting can successfully deal with outliers, but when it is not dominant, fitting can fail. However, some failures of fitting in a frame can be fixed later by the non-causal TbD (TbD-NC).

Figure 5.11. Failures due to a false positive of the FMO detector. From left to right: input image region where FMOd incorrectly indicated a hit, background estimated by moving median, estimated blur H, and trajectory fit that passed the consistency check with H.

TbD-NC is limited by our assumption that the object motion under all forces can be approximated by a piecewise polynomial function. For instance, if somebody ties up a ball on a rope and rotates it or makes arbitrary motion with it, splitting into segments can fail. Another example is object motion under the influence of fans.

False negatives occur, for example when objects fly over saturated background, collide with other moving objects, or get occluded (missed bounce in "air hockey" Figure 5.9 partially hidden by the frame bottom edge).

Figure 5.10 contains examples of failed blur estimation (and subsequently failed trajectory estimation) due to discrepancy in the estimated and actual background image B. The deblurring step is quite sensitive to having the correct background as input and when another object gets close enough to appear in the domain D (typically one of the players with their hand or racket, or some moving object in the far background, as in the case on the second row in Figure 5.10), the background estimation by moving median fails and as a result also the deblurring and trajectory estimation are likely to fail.

Figure 5.11 shows different kind of failures, false positives. When the object is lost (or initially for the first detection), the FMO detector from [RKŠ+17] is used to detect a candidate for tracking. Sometimes FMOd gives a false positive in an area with some background motion but without the object of interest and this false positive passes through the TbD pipeline, and the corresponding estimated blur is classified as a motion trajectory. As a result, TbD starts tracking an entirely different object. Figure 5.11 shows examples of several of such cases.

The TbD method assumes that the exposure fraction is given, which is usually static for the whole sequence. However, there are cameras with dynamic exposure fraction. It is possible to include dynamic exposure fraction estimation in the TbD framework, which will include only several previous frames to calculate the exposure fraction in the current frame. However, all

sequences in the FMO dataset and in the TbD dataset have constant exposure fraction.

There is also some ambiguity in the Trajectory-IoU measure. For instance, two cases of "similar mediocre accuracy over the whole time" and "very accurate half the time, and then lost with zero accuracy" might have exactly the same TIoU score. To disambiguate these two cases we could report not only the mean TIoU but also its standard deviation σ. Low standard deviation implies similar accuracy over all time and the second situation will be indicated by high σ. However, in the experiments we avoided this due to the negligible influence on the comparison. Even with the explained ambiguity of TIoU, this is still a better score measure than the standard IoU measure, which will give even more uncertainty.

5.8. Settings

We used the following L1 weight on H in deblatting: $\alpha_H = 2$, and for sequences badminton white, badminton yellow, pingpong and throw soft it was set to $\alpha_H = 0.2$ due to low contrast. The norm on F was set to $\alpha_F = 2^{-10}$ for all sequences. The threshold for Consistency Check τ was set to 0.5 everywhere. We fixed template-matching term λ to 0.1. For speed-up, some sequences were downscaled.

The running time per frame of TbD depends on the ROI size (D) in the deblurring step. The ROI dimensions are calculated in the motion prediction step and depend on the size of the tracked object (M) and predicted trajectory length ($|\mathcal{C}_t|$). For the presented TbD dataset, the average ROI size was 100×150 pixels and we achieved 0.5 fps in Matlab on the 6th generation CPU Intel Core i7.

CHAPTER 6
CONCLUSIONS

Small objects moving along complex trajectories with varying speed is a common phenomenon in real-life videos, especially sports. Tracking such objects is considerably different from standard object tracking targeted by state-of-the-art algorithms. We proposed a novel approach of Tracking by Deblatting, *debl*urring and m*atting*, based on a notion that motion blur in frames is directly related to object trajectories and by estimating the blur, objects are precisely localised in time. The method can track objects travelling at a wide range of speeds and without a priori knowing their appearance. The estimated trajectories have temporal resolution much higher than a traditional one sample per frame.

Tracking by Deblatting is intended for sequences in which the object of interest undergoes non-negligible motion within a single frame which needs to be specified by intra-frame trajectory rather than a single position. The blur is estimated by a complex method combining blind deblurring, image matting and shape estimation, followed by fitting a piecewise linear or quadratic curve that models physically plausible trajectories. As a result, we can precisely localise the object with higher temporal resolution than by conventional trackers.

The non-causal Tracking by Deblatting (TbD-NC) estimates more accurate and complete trajectories than the causal TbD. TbD-NC is based on globally minimising an optimality condition which is done by dynamic programming. High-order polynomials are then fitted to trajectory segments without bounces. The final output is a continuous trajectory function which assigns location for every real-valued time stamp from zero to the number of frames.

The proposed TbD method was evaluated on a newly created dataset of videos with ground truth obtained by a high-speed camera using a novel Trajectory-IoU metric that generalises the traditional Intersection over Union and measures the accuracy of the intra-frame trajectory. The TbD method outperforms baseline techniques by a wide margin both in recall and trajectory accuracy. The non-causal TbD-NC method performs even better and complete failures on the TbD dataset appear 10 times less often than for the causal TbD method. From the estimated trajectories, we are able to calculate precise object properties such as velocity or shape. The speed estimation is compared to the data obtained from a high-speed camera and radar guns. Applications such as fast moving objects removal and temporal super-resolution are shown.

Due to the complexity of blind deblurring, the method is currently limited to objects that do not significantly change their perceived shape and appearance within a single frame, the method works best for approximately round and uniform objects.

BIBLIOGRAPHY

[Avi07] Shai Avidan. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):261–271, February 2007. 5

[B+00] Margrit Betke et al. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications*, 12(2):69–83, 2000. 1

[BCR15] Tewodros A. Biresaw, Andrea Cavallaro, and Carlo S. Regazzoni. Correlation-based self-correcting tracking. *Neurocomput.*, 152(C):345–358, March 2015. 5

[BYB11] B. Babenko, M. H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, Aug 2011. 5

[CRM03] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, May 2003. 5

[DHSKF14] Martin Danelljan, Gustav Hauml;ger, Fahad Shahbaz Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014. 1, 5

[DHSKF15] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015. 5

[DW08] Shengyang Dai and Ying Wu. Motion from blur. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, June 2008. 5

[F+00] SN Fry et al. Tracking of flying insects using pan-tilt cameras. *Journal of Neuroscience Methods*, 101(1):59–67, 2000. 1

[GRB13] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Comput. Vis. Image Underst.*, 117(10):1245–1256, October 2013. 5

[HGS+16] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2096–2109, Oct 2016. 5

[Hra17] Aleš Hrabalík. Implementing and applying fast moving object detection on mobile devices, master's thesis. *Czech Technical University in Prague, Faculty of Electrical Engineering*, 2017. 3, 6, 35

[J+18] M. Jin et al. Learning to extract a video sequence from a single motion-blurred image. In *IEEE CVPR*, pages 6334–6342, June 2018. 5

[Jia07] Jiaya Jia. Single image motion deblurring using transparency. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8, 2007. 5

[K+16] Matej Kristan et al. *The Visual Object Tracking VOT2016 Challenge Results*, pages 777–823. Springer International Publishing, Cham, 2016. 1, 5

[K+19] Matej Kristan et al. The sixth visual object tracking vot2018 challenge results. In Laura Leal-Taixé and Stefan Roth, editors, *ECCV 2018 Workshops*, pages 3–53, Cham, 2019. Springer International Publishing. 1, 25, 29

[KDVG14] Till Kroeger, Ralf Dragon, and Luc Van Gool. Multi-view tracking of multiple targets with dynamic cameras. In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *Pattern Recognition*, pages 653–665, Cham, 2014. Springer International Publishing. 1

[KL14] Tae Hyun Kim and Kyoung Mu Lee. Segmentation-free dynamic scene deblurring. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2766–2773, 2014. 5

[KML+15] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015. 1

[KML+16a] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, Nov 2016. 1

[KML+16b] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers, Jan 2016. 1

[KMM12] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012. 1

[KŠ18] J. Kotera and F. Šroubek. Motion estimation and deblurring of fast moving objects. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2860–2864, Oct 2018. 6

[LČZV+18] Alan Lukežič, Luka Čehovin Zajc, Tom'aš Voj'iř, Jiř'i Matas, and Matej Kristan. Fucolot - a fully-correlational long-term tracker. In *ACCV*, 2018. 2, 5, 9, 25, 29, 32, 33

[LLW08] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, February 2008. 5

[LVC+17] Alan Lukezic, Tomas Vojir, Luka Cehovin, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 5, 25, 29, 32

[M+16a] B. Ma et al. Visual tracking under motion blur. *IEEE TIP*, 25(12):5867–5876, December 2016. 5

[M+16b] Matthias Mueller et al. A benchmark and simulator for uav tracking. In *ECCV*, pages 445–461, 2016. 1

[MD03] Anurag Mittal and Larry S Davis. M 2 tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV*, 51(3):189–203, 2003. 1

[MG17] Abhinav Moudgil and Vineet Gandhi. Long-term visual object tracking benchmark. *arXiv preprint arXiv:1712.01358*, 2017. 1

[R+16] Ergys Ristani et al. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV 2016 Workshops*, pages 17–35, 2016. 1

[RKŠ+17] Denys Rozumnyi, Jan Kotera, Filip Šroubek, Lukas Novotny, and Jiri Matas. The world of fast moving objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3, 5, 6, 8, 18, 25, 29, 30, 31, 32, 33, 40

[Roz17] Denys Rozumnyi. Tracking, learning and detection over a large range of speeds, bachelor thesis. *Czech Technical University in Prague, Faculty of Electrical Engineering*, 2017. 1, 2, 3, 6

[RT18] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[S+17a] Clemens Seibold et al. Model-based motion blur estimation for the improvement of motion tracking. *CVIU*, 160:45–56, 2017. 5

[S+17b] S. Su et al. Deep video deblurring for hand-held cameras. In *IEEE CVPR*, pages 237–246, July 2017. 5

[SCXP15] Jian Sun, Wenfei Cao, Zongben Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 769–777, 2015. 5

[SXJ07] Qi Shan, Wei Xiong, and Jiaya Jia. Rotational motion deblurring of a rigid object from a single image. In *Proc. IEEE 11th International Conference on Computer Vision ICCV 2007*, pages 1–8, October 2007. 5

[T+17] Ran Tao et al. Tracking for half an hour. *arXiv preprint arXiv:1711.10217*, 2017. 1

[TK91] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991. 5

[TYZW18] Ming Tang, Bin Yu, Fan Zhang, and Jinqiao Wang. High-speed tracking with multi-kernel correlation filters. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 5

[VNM13] Tomas Vojir, Jana Noskova, and Jiri Matas. *Robust Scale-Adaptive Mean-Shift for Tracking*, pages 652–663. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. 1, 5

[W+11] Y. Wu et al. Blurred target tracking by blur-driven tracker. In *IEEE ICCV*, pages 1100–1107, November 2011. 5

[W+17] P. Wieschollek et al. Learning blind motion deblurring. In *IEEE ICCV*, pages 231–240, Oct 2017. 5

[WLY13] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1

[ZMS14] Jianming Zhang, Shugao Ma, and Stan Sclaroff. *MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization*, pages 188–203. Springer International Publishing, Cham, 2014. 5

APPENDIX A

CD CONTENT

```
/
├── thesis ..... LaTeX source code for the thesis
│   └── thesis.pdf ..... Compiled thesis
├── data ..... Used data
├── src ..... Implementation of TbD and TbD-NC
│   └── go.m ..... Examples to run the code
└── demo ... Videos of trajectory estimation, FMO removal and temporal super-resolution
```