CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

BACHELOR THESIS

ISSN 1213-2365

Tracking, Learning and Detection over a Large Range of Speeds

Denys Rozumnyi

rozumden@fel.cvut.cz

26 May 2017

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

This research has been supported by the Technology Agency of the Czech Republic research program TE01020415 (V3C – Visual Computing Competence Center) and the Grant Agency of the CTU Prague under Project SGS17/185/OHK3/3T/13.

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

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Denys R o z u m n y i

Study programme: Open Informatics

Specialisation: Computer and Information Science

Title of Bachelor Project: Tracking, Learning and Detection over a Large Range of Speeds

Guidelines:

1. In [1], the authors proposed a method for tracking very fast moving objects and showed that standard state-of-the-art trackers fail for such objects.
2. On the other hand, the FMO method proposed in [1] fails on objects that are not moving fast enough to display significant blur.
3. Familiarise yourself with the state-of-the-art in standard object tracking methods [2].
4. Propose an algorithm combining a standard, "slow motion" tracker and the FMO method [1] to achieve tracking in a wide range of speeds. Aim at a long-term tracker that is able to learn and re-detect a temporarily lost object.
5. Implement the method and evaluate its performance.

Bibliography/Sources:

[1] Denys Rozumnyi et al.: The World of Fast Moving Objects. In arXiv volume 1611.07889, November 2016.
[2] Matej Kristan et al.: The Visual Object Tracking VOT2015 challenge results. In The IEEE International Conference on Computer Vision (ICCV) Workshops, December 2015.

Bachelor Project Supervisor: prof. Ing. Jiří Matas, Ph.D.

Valid until: the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 6, 2017

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

Abstrakt

V této práci navrhujeme algoritmus pro detekci a sledování objektů, které se v záběru kamery pohybují vysokou rychlostí, nicméně mohou se i zpomalit. Objekt nazýváme rychle se pohybujícím (vůči nějaké kameře), jestli jeho trajektorie po dobu jednoho snímku překročí jeho velikost. Takové objekty jsou často špatně viditelné a vypadají jako poloprůhledné pruhy. První, detekční část navrhovaného algoritmu je schopná nalézt rychle se pohybující objekty bez předchozích znalostí. Druhá část algoritmu slouží pro dlouhodobé sledování, a zvládá nepřetržité sledování i těch objektů, které zpomalí a již nejsou rychle se pohybující.

Pro vyhodnocení algoritmu jsme připravili datovou sadu FMOv2. Výsledky ukazují, že navržená metoda překonává dosavadní algoritmy sledování, pokud jsou objekty rychle se pohybující. Také jsme předvedli několik aplikací detekce a sledování rychle se pohybjících objektů, například zvýraznění sledovaného objektů, měření rychlosti, časové super rozlišení, a jiné.

Klíčová slova: vizuální sledování objektů, rychle se pohybující objekty, časové super rozlišení

Abstract

In this thesis we propose an algorithm which allows detection and tracking of objects that appear in videos as fast moving which can possibly slow down. Object is fast moving (with respect to a camera) if its projected trajectory is larger than its size in one frame. In a single frame, such objects are often barely visible and appear as semi-transparent streaks. The detection part of the algorithm can discover previously unseen fast moving objects. The long-term tracking part is able to continuously track objects even when they are no longer fast moving.

For the method evaluation we introduce FMOv2 dataset. The results show that the proposed method outperforms existing trackers when objects are fast moving. We demonstrate several applications of fast moving object detection and long-term tracking, such as temporal super-resolution, highlighting, speed estimation and other.

Keywords: visual object tracking, fast moving objects, temporal super-resolution

Acknowledgements

I would like to thank my thesis supervisor prof. Jiří Matas for his excellent guidance throughout this work. His professional expertise helped me to keep this research in the right direction. Large amount of this work was done during the collaboration with Filip Šroubek and Jan Kotera and their help is truly appreciated. The dataset was acquired in cooperation with Aleš Hrabalík, Jan Kotera, Filip Šroubek and Lukáš Novotný.

I would like to express my gratitude to my colleagues at CMP who I had the pleasure to work with, especially James Pritts and the whole G3 team. I thank Filip Radenović for his important comments that allowed me to improve the quality of the thesis.

Also, I am indebted to my family and friends who provided me with support. This accomplishment would not be possible without them.

Contents

List of Figures

List of Tables

Term	Description
t	frame number (time)
I_t	image frame at time t
G_t	gradient magnitude of image frame at time t
O_t	gradient orientation of image frame at time t
$\mathcal{A}_t$	affine transformation between adjacent image frames
ε	exposure fraction
B	background
F	fast moving object model
M	indicator function of F
$\mathcal{H}$	blur operator
$[\mathcal{H}M]$	object visibility map
$[\mathcal{H}F]$	blurred object appearance
P	object trajectory (path)
a	object axis of rotation
ϕ	object angle of rotation
r	object radius
L	fitted curve (line) for the object trajectory
A	area of the object
Δ	temporal difference image
θ	threshold for temporal difference image
i	offset for the discrete computation of derivative
s	percentage of highest pixels in Δ which contain FMO
D	distance transform of the temporal difference image
LM	local maxima of the distance transform
O_t	FMOs at time t
o	a single FMO
$\mathcal{V}$	all FMO models
β_i	inlier threshold for model assignment
β_r	threshold for trajectory pixels
β_f	threshold for maximal value of derivative
β_s	threshold for maximal allowed stroke area ratio
β_g	maximal allowed gradient magnitude difference
β_o	maximal allowed gradient orientation difference

Table 0.1. The most commonly used denotations in the thesis.

Abbreviation	Meaning
FMO	Fast Moving Object
SMO	Slow Moving Object
VOT	Visual Object Tracking
CVPR	Computer Vision and Pattern Recognition
PoC	Proof-of-Concept
CSR-DCF	Discriminative Correlation Filter with Channel and Spatial Reliability
SRDCF	Spatially Regularised Discriminative Correlation Filters
ALOV	Amsterdam Library of Ordinary Videos
OTB	Online Tracking Benchmark
GPU	Graphics Processing Unit
FREAK	Fast REtinA Keypoint
FAST	Features from Accelerated Segment Test
CC	Connected Component
RANSAC	RAndom SAmple Consensus
LO-RANSAC	Locally Optimised RAndom SAmple Consensus
SVM	Support Vector Machine
HoG	Histogram-of-Gradients
KLT	Kanade-Lucas-Tomasi tracker
MATLAB	MATrix LABoratory
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
IoU	Intersection over Union
ASMS	Adaptive Scale Mean-Shift tracker
DSST	Discriminative Scale Space Tracker
STRUCK	Structured Output Tracking with Kernels
MEEM	Multiple Experts using Entropy Minimisation tracker
fps	frames per second

Table 0.2. Abbreviation table.

1. Introduction

Visual object tracking (VOT) is an important problem in the computer vision field. It has received an enormous attention of the research community. Methods for object tracking based on various principles have been proposed and many surveys have been conducted [2, 3, 4]. For example, around 100 trackers were tested only during the VOT 2016 challenge [5].

Visual object tracking is a broad term and it covers many different problems. In the basic definition VOT tries to continuously establish point-to-point correspondences of the query object in image frames. Alternatively, VOT can be formulated as a segmentation of the object in image sequences.

Methods based on VOT have many applications, which are important in real-world situations, such as human computer interaction, augmented reality, management of video content (indexing & search), film production and post-production, action and activity recognition, assistance, surveillance, defence, robotics, autonomous car driving, medicine measurements and others.

Recently, in [1] we proposed to study one of the sub-problems of VOT – phenomena that appear in videos and images as fast moving objects (FMOs). We use the term FMO for objects whose projected trajectory is larger than its size in a single frame. Consequently, FMO is a property of a camera and an object. Figure 1.1 shows several examples of how FMOs can look like.

Detection, tracking and appearance reconstruction of FMOs allow performing tasks with applications in diverse areas. In addition to many applications which are common for all tracking methods, we show the ability to synthesise realistic videos with higher frame rates (temporal super-resolution), artificial object highlighting or deletion, visualisation of rotational axis and measurement of speed and angular velocity. The extracted properties of FMOs, such as trajectory, rotation angle and velocity have applications, e.g. in sports analytics. If the object appearance model is of interest, the shape and deformations during the object motion, that are otherwise invisible, can be recovered. FMOs are also essential in mechanics – blur gives us information about fast vibrating objects; in scanning probe microscopy – blur encodes the mea-

Figure 1.1. Examples of fast moving objects that appear as semi-transparent streaks: (left-to-right, top-to-bottom) table tennis, archery, volleyball, tennis, hailstorm and flying insects.

suring tip shape and scanning procedure; in ophthalmology – estimated blur in retinal images carries information about pupil abnormalities; in forensic science – blur in consecutive video frames provides hints, such as exposure fraction, for identification of the capturing sensor.

The problem of fast moving objects had not been addressed by the computer vision community and we have introduced it in [1] for the first time. We can speculate about the reasons why FMOs have not been studied before, but for most people and researchers the following is still true – if the image (video) is blurred, it is better to re-capture it. However, images and videos of objects moving fast with respect to the camera always contain a significant amount of blur. Blur is commonly considered a nuisance – it is a typical reason for computer vision algorithms to fail – and yet it encodes important information about motions and the sensor. Large blur complicates object tracking, which is arguably the reason why none of the current tracking techniques and benchmarks include objects that are moving so fast to appear as streaks. This is a surprising omission considering the fact that such objects are common in real-world situations, in which sports play a prominent role. We have shown that detection and tracking of such objects in videos is feasible, at least in some restricted scenarios [1]. This makes the investigation of FMOs realistic, while opening a range of related basic research problems.

1.1. Contributions

This thesis extends the Proof-of-Concept (PoC) algorithm [1] in many important ways. The PoC algorithm has several assumptions which lead to a low precision rate. Additionally, it completely fails on objects which are not moving fast enough to display significant blur. This is unacceptable when the full analysis of the object and its trajectory is required. For example, quite often FMOs slow down and become slow moving objects (SMOs) or vice versa. Tracking of objects which alter their state of being FMO and SMO can be considered as long-term FMO tracking. The main goal of this thesis is to develop such a method which is capable of long-term FMO tracking. The reason why it is not considered as long-term tracking is because when the object is temporarily lost, it can be re-localised again only as an FMO. The problem of long-term FMO tracking is not easy, because FMO detection and tracking fail on slow moving objects and standard tracking methods fail on FMOs. The thesis has 3 main contributions:

- A new method for continuous tracking of objects that are no longer FMOs is introduced. The proposed method, names long-term FMO-SMO tracking, combines FMO detection with the state-of-the-art CSR-DCF tracker [6].
- The FMO dataset has been extended. We also included videos with multiple FMOs or additionally provided annotations where the second, usually smaller FMO, was ignored. Ground truth is now available even when FMOs slow down and do not contain enough blur, which means that they are no longer FMOs. We denote this dataset FMOv2 and it can be considered as a meeting point of traditional tracking and FMO tracking.
- As part of the proposed long-term tracker, we introduced a new very precise algorithm for improved FMO detection which is based on the analysis of connected components in temporal difference images. It has average precision near 93% compared to 55.7% average precision of PoC algorithm on the FMO dataset [1].

1.2. Thesis structure

Chapter 2 discusses the related work on the topic. The problem formulation is defined in chapter 3. Robust FMO detection is introduced in chapter 4. Next, chapter 5 explains a method for long-term FMO-SMO tracking. Evaluation is conducted in chapter 6. Finally, we conclude the thesis in chapter 7.

2. Related work

Besides the Proof-of-Concept algorithm [1], there is no literature on fast moving objects. This makes the investigation of related work difficult. The literature sometimes refer to fast moving objects, but only the case with no significant blur is considered [7, 8, 9]. Thus, in this chapter we will focus on standard "slow" tracking methods.

Methods for object tracking based on various principles have been proposed and several surveys have been compiled [2, 3, 4]. A range of methods has been proposed based on diverse principles, such as correlation [10, 11, 12, 6], feature point tracking [13], mean-shift [14, 15], and tracking-by-detection [16, 17]. As we show in the evaluation section, none of the methods is able to handle a large amount of blur.

Visual object tracking has shown excellent performance of discriminative correlation filters [10, 11, 12, 6]. Recently, Lukežič *et al*. proposed a new correlation-based tracker – CSR-DCF [6], which achieves state-of-the-art results on standard tracking datasets. We have chosen this tracker and integrated it to the proposed long-term FMO-SMO tracking method. Apart from the fact that CSR-DCF has state-of-the-art performance, it is fast, simple, easy to implement and has no GPU requirements.

Standard benchmarks, some comprising hundreds of videos, such as ALOV [18], VOT [19, 5] and OTB [20], are available. Yet none of them includes objects that are moving so fast that they appear as streaks much larger than their size – with significant blur and large frame-to-frame displacement. We have analysed them and compared to the FMOv2 dataset in terms of the motion of the object of interest. For example, in the conventional datasets, the object frame-to-frame displacement is below 10 pixels in 91% of cases, while in the FMOv2 dataset the displacement is almost uniformly spread between 0 and 150 pixels. Similarly, the intersection over union (IoU) of bounding boxes between adjacent frames is above 0.5 in 94% of times for the conventional datasets, whereas the proposed dataset has zero intersection nearly every time. Figure 2.1 summarises these findings.

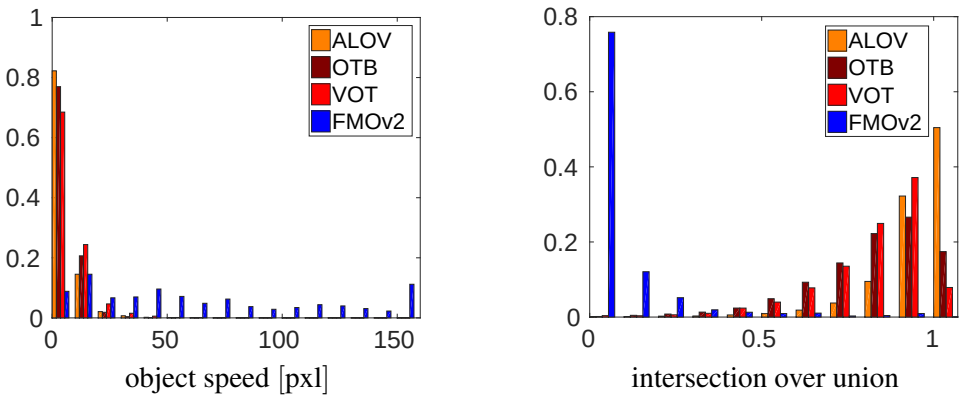

Figure 2.1. The FMOv2 dataset includes motions that are an order of magnitude faster than three standard datasets - ALOV, VOT, OTB [18, 19, 20]. The figure illustrates normalised histograms of projected object speeds (left) and intersection over union IoU of bounding boxes (right) between adjacent frames.

3. Problem formulation

Definition 1. Fast Moving Object (FMO). Object is fast moving (with respect to a camera) if its trajectory projected on the image plane is larger than its size in a single frame. FMO can also rotate along an arbitrary axis with an unknown angular speed.

3.1. Camera exposure

Two parameters are relevant when capturing videos - frame rate and exposure time. Frame rate, measured in frames-per-second (fps), is the frequency at which frames are captured. Current video cameras usually have 25-30 fps frame rate, while 250-300 frame rates are also quite common (e.g. iPhone). Though in some area, such as film producing, video cameras with much larger rates are used – up to 21,500 fps (e.g. Chronos 1.4).

The second parameter – exposure time, or shutter speed, τ [in seconds] defines the duration for which the camera sensor is open and captures the incoming light. This camera parameter determines the amount of blur present in video frames. The exposure time must be shorter than frame duration or equal to it, thus $\tau \leq 1/\text{fps}$. Frame duration plays the role of the upper bound for the exposure time.

Frame rate and exposure time are global parameters for the camera and cannot be determined only from the captured video, unless the real object size or speed is known. The common solution is the information provided directly by the camera in the video file. If none of them are available, we can define the exposure fraction $\varepsilon = \tau \times \text{fps}$ which is the fraction of the time when the sensor is open. As we further demonstrate, the proposed method can determine the exposure fraction solely from the fast moving objects. Then, if the frame rate is known, the exposure time can be calculated as $\tau = \varepsilon/\text{fps}$ seconds.

3.2. Image formation in the presence of FMO

For simplicity, we assume a single object F moving over a static background B; an extension to multiple objects is relatively straightforward. To get close to the static background state, camera motion is assumed to be compensated by video stabilisation, which will be discussed in section 3.5.

Let a recorded video sequence consist of frames $I_1(x), \ldots, I_n(x)$, where $x \in \mathbb{R}^2$ is a pixel coordinate. Frame I_t is then formed as

$$I_t(x) = \Big(1 - [\mathcal{H}_t M]\,(x)\Big) B_t(x) + [\mathcal{H}_t F](x)\,, \tag{3.1}$$

where M is the indicator function of F. The indicator function, or also the characteristic function, has the value 1 for all x which are members of F and 0 otherwise.

In general, the operator $\mathcal{H}_t$ models the blur caused by object motion and rotation, and performs the 3D→2D projection of the object representation F onto the image plane. We assume that speed of the object is nearly constant during the exposure time of a single frame (usually less than $1/25 = 0.04$ seconds). Then, this operator depends mainly on three parameters, $\{P_t, a_t, \phi_t\}$, which are the FMO trajectory (path), and the axis and angle of rotation, respectively.

The $[\mathcal{H}_t M](x)$ function corresponds to the object visibility map (alpha matte, relative duration of object presence during exposure) and appears in (3.1) to merge the blurred object and the partially visible background. The $[\mathcal{H}_t F](x)$ function represents the blurred appearance of the object.

The object trajectory P_t can be represented in the image plane as a path (set of pixels) along which the object moves during the frame exposure. In the case of no rotation or when F is homogeneous, i.e. the surface is uniform and thus rotation is not perceivable, $\mathcal{H}_t$ simplifies to a convolution in the image plane:

$$[\mathcal{H}_t F](x) = \frac{1}{|P_t|}[P_t * F](x), \tag{3.2}$$

where $|P_t|$ is the path length, F can then be viewed as a 2D image. Let us define a temporal difference image as the L_1 norm of every pixel differences:

$$\Delta_t = \|I_t - B_t\|_1. \tag{3.3}$$

Substituting I_t by its formation model (3.1) gives:

$$\Delta_t = \Big\|\big(1 - [\mathcal{H}_t M]\big)B_t + [\mathcal{H}_t F] - B_t\Big\|_1 = \Big\|[\mathcal{H}_t F] - [\mathcal{H}_t M]B_t\Big\|_1. \tag{3.4}$$

Thus, in the presence of FMO temporal difference image Δ_t is non-zero when there is some contrast between FMO and the background. Also, the response in Δ_t is largely determined by FMO speed – the faster object is, the more background dominates in the model formation equation (3.1).

3.3. Background modelling

The formation model (3.1) is a complex equation and we need to simplify it in order to solve the problem (i.e. find FMOs). The first step was to assume homogeneous sphere or no rotation (equation (3.2)) and consider nearly constant speed during the exposure time of a single frame. But everything becomes even simpler, if we have the background B_t. In some situations the background can be considered as a previous frame, i.e. $\hat{B}_t = I_{t-1}$. However, the difference image found using the previous frame will combine FMOs in frames I_t and I_{t-1}.

Based on the FMO property that it elapses distance exceeding its size within one frame, the background can be also calculated by taking median of previous 3 or more frames:

$$\hat{B}_t = \text{median}\{I_t, I_{t-1}, I_{t-2}, \dots\}. \tag{3.5}$$

This operation is computationally cheap and provides background approximation enough for the temporal difference image in equation (3.3), giving $\hat{\Delta}_t = |I_t - \hat{B}_t|$. In case of a full exposure camera due to a small overlap of adjacent FMOs, background from the median of 3 frames will lead to small errors at this areas. However, the median of 5 frames will solve this problem. Figure 3.1 illustrates an example of the found background from 5 frames on a sequence with 5 table tennis balls. Clearly, the background approximation does not have any FMOs.

Finding all the intrinsic and extrinsic properties of arbitrary FMOs means estimating both F and $\mathcal{H}_t$, which is, at this moment, a difficult task. To alleviate this problem, some prior knowledge of F is necessary. In our case, the prior is in the form of object shape. Since in most sport videos the FMOs are spheres (balls), we continue our theoretical analysis focusing on spherical objects. Although, when an object moves very fast and rotates with a high angular speed, which is common for FMOs, it can be approximated by a sphere. As we further demonstrate, the proposed localisation method can successfully handle objects of significantly different shapes.

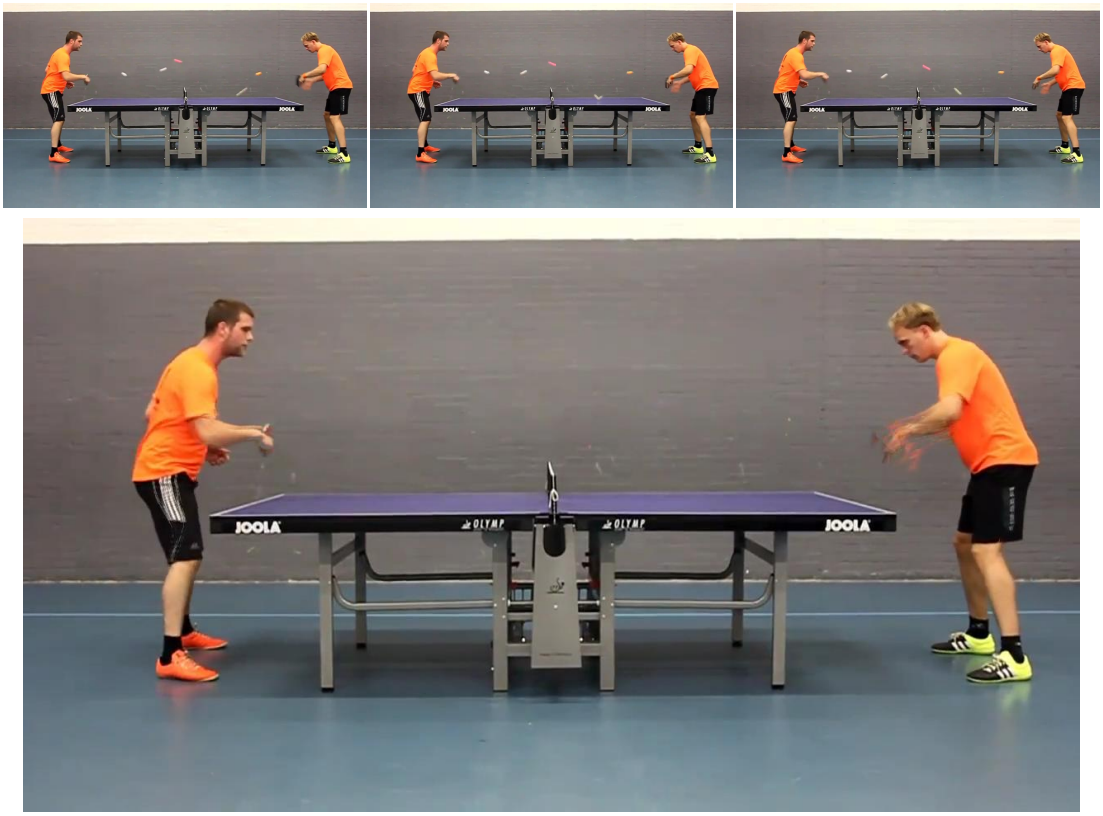

Figure 3.1. Example of the background estimation on a sequence with 5 table tennis balls that are FMOs. Median of 5 consecutive frames (3 of them are in the top row) gives the background B_t (the bottom image). The players are moving but their motion is not interpreted as FMO (see Algorithm 1).

3.4. FMO localisation

In classical object tracking, the initial position of the object of interest is given. However, in the case of FMOs, the initial location can be established automatically. Thus, we propose a method for efficient and reliable FMO localisation, i.e. detection and tracking. FMO detector can explore new unseen fast moving objects and it requires only image frames as input. FMO tracker additionally requires properties and location of the tracked fast moving objects.

If some FMO is localised at least twice in the consecutive frames, the exposure fraction ε can be established by the ratio of the FMO length to the distance between the start points of the trajectories of each FMO.

The FMO appearance, and the axis and angle of the object rotation, can be reconstructed which requires the precise output of the method. The tracker output (trajectory P_t and radius r) then can be used to initialise the precise estimation of appearance using the full model (3.1).

3.5. Camera stabilisation

The proposed FMO detection and tracking require a static background or a registration of consecutive frames. We assume that camera stabilisation is performed before the algorithms are run. Camera stabilisation is a well-studied topic and many approaches are available. We need a camera stabilisation method which is as fast as FMO detection and tracking, while providing pixel-to-pixel accuracy. For this task, feature based methods are known to be robust [21].

The transformation between image frames is determined by a homography matrix which models the isomorphic function between projective image spaces. We tried to calculate homography but because camera does not usually move that much, a rough estimate by an affine matrix provides sufficient approximation with a sub-pixel accuracy.

In order to estimate a transformation between frames, local features are used to extract high-level representations of the image. We use robust FAST corner features [22], which are known to be invariant to translation, reflection and rotation. We expect small changes of the view and these invariants are sufficient. More importantly, FAST features, as their name suggests, are computationally efficient and their extraction is faster than many other image features.

To this end, we apply video stabilisation by estimating the affine transformation between frames. FREAK descriptors [23] of FAST features [22] provide tentative matches. Then, the model with the highest number of inliers is found using LO-RANSAC [24]. For example, if we denote an affine transformation from the frame $t-1$ to t as $\mathcal{A}_t$, then the background approximation with dynamic camera will be modified to:

$$\hat{B}_t^{\,d} = \text{median}\{I_t, \mathcal{A}_t \odot I_{t-1}, \mathcal{A}_t \odot \mathcal{A}_{t-1} \odot I_{t-2}, \dots\}, \tag{3.6}$$

where $\mathcal{A}_t \odot I_{t-1}$ denotes applying affine transformation $\mathcal{A}_t$ on image I_{t-1}. The FMO detection and tracking is currently not robust to incorrect stabilisation.

4. FMO detector

4.1. Initial observations

FMOs are present in temporal difference images. This observation is a consequence of the equation (3.4), where $\Delta_t = \|[\mathcal{H}_t F] - [\mathcal{H}_t M] B_t\|_1$. It is clear that in most situations fast moving objects will be present in temporal difference images. However, the equation will be zero at pixels where the background is similar to FMO. Thus, we assume that FMOs have enough contrast compared to the background in order to be detected.

FMOs are connected components (CC). Connected components are extracted from binarised temporal difference images $\Delta_t^b = \Delta_t > \theta$. Connected components are defined as maximal sets of positive pixels which have paths between every pair of pixels in 8-pixel neighbourhood (or each of the connected components has no connection with other components in 8-pixel neighbourhood). The assumption that FMOs are CC simplifies the search space and almost does not affect the performance. FMOs are not connected components only when there is a low contrast between the background and FMO in some regions, which brakes FMO into several parts.

FMOs are strokes. Strokes are regions for which the following applies: its boundary pixels are equidistant from its main axis (or trajectory) and the axis is a curve (for simplicity we assume a straight line). FMO formation model (3.1) guarantees that it is true when FMO is a sphere. For different shapes it will hold in case when the trajectory is long enough. Another example of strokes are text characters [25].

FMOs have different properties than other strokes. Based on preliminary experiments we established that *connected components* in *temporal difference images* which are *strokes* appear due to 4 different physical phenomena: lateral motion of an object with long low-curvature edge, fast change of illumination on regions with stripe-like patterns, or on the edge of shadows. The last phenomenon is FMO, which we are interested in. The gradient fields of the current and the previous frame give enough discriminative information to distinguish between FMOs and 3 other types of strokes.

All the observations stated above are the grounds for an algorithm for FMO detection, which will be explained in the following sections.

4.2. The detection algorithm

The detector is the only generic algorithm for FMO localisation that requires no input, except for image frames. Thus, it has the ability to discover fast moving objects. Consequently, it is the most important part in FMO localisation pipeline – if the object cannot be explored, it cannot be tracked either. Needless to say, the detector should be very precise in exploring fast moving objects.

One of the drawbacks of the PoC (Proof-of-Concept) algorithm [1] is that it requires at least three appearances of an FMO in order to find it. In consequence, objects that cross the field of view in one or two frames, a common situation for very rapid movements, are not detected and hence not tracked. The proposed algorithm is able to detect FMOs in a single frame. In order to achieve this, we also provide two improvements for calculating temporal difference images. First, the background is established by taking median of previous frames. Second, the threshold for Δ_t is adaptively estimated for different images.

Algorithm 1 shows pseudo-code which ignores some technical details and gives a high level insight into FMO detector. Each step of the algorithm is pictorially visualised in Figure 4.2. Further detailed descriptions are in the following sections.

Algorithm 1 FMO detector

1: $B_t \leftarrow \text{median}\{I_t, I_{t-1}, I_{t-2}\}$
2: Estimate θ^* ▷ see section 4.3
3: $\Delta_t^b \leftarrow \|I_t - B_t\|_1 > \theta^*$
4: **for** CC $\in$ connectedComponents(Δ_t^b) **do** ▷ for every connected component in Δ_t^b
5: **if** CC is a stroke ▷ see section 4.4
6: & CC is not a shadow or illumination change ▷ see section 4.5.2
7: & CC is not a lateral motion **then** ▷ see section 4.5.1
8: CC is FMO
9: **end if**
10: **end for**

The ordering of operations at lines 5-7 in the Algorithm 1 has a crucial impact on the speed. Nevertheless, the permutation of these operations would not change the performance. Based on the preliminary experiments, we established that most of the connected components are not *strokes*, while just some of them are *shadows or illumination changes* or *lateral motion of objects*. The reason why *shadow or illumination change* test is performed before the *lateral motion* test is because the latter is computationally more expensive (though still fast).

4.3. Threshold estimation

Image noise can seriously affect quality of temporal difference images Δ_t. There are numerous potential sources of noise, such as photon noise, readout noise, dark noise [26] or noise caused by imperfect camera stabilisation. Thus, the parameter θ for binarised equation (3.3) must be adaptive and be able to handle variate levels of image noise. Figure 4.1 shows how different thresholds can change the temporal difference images and highlights the need for an adaptive method for the threshold estimation. If the threshold is too small, then connected components which are close to each other have a large probability to be connected by noise. Whereas, too large threshold can entirely remove FMOs from the difference images.

Let us define the normalised image histogram of Δ_t as $P(\Delta_t = \theta) = P(\theta)$, which is equal to a relative likelihood that any randomly chosen value in Δ_t is θ. In the case of a discrete set of values in Δ_t, which is in our case values from 0 to 255, the histogram can be calculated by number of occurrences of the given θ divided by the number of pixels. Also, we define the cumulative histograms as $C(\Delta_t = \theta) = C(\theta) = \sum_{\theta_i \leq \theta} P(\theta_i)$. Both histograms can be seen as a probability function or a cumulative probability function, respectively. Figure 4.3 demonstrates how histograms for Δ_t can look like. We estimate the threshold θ^* by exploring properties of the image histogram for every frame. The final threshold is equal to the maximum of two following thresholds.

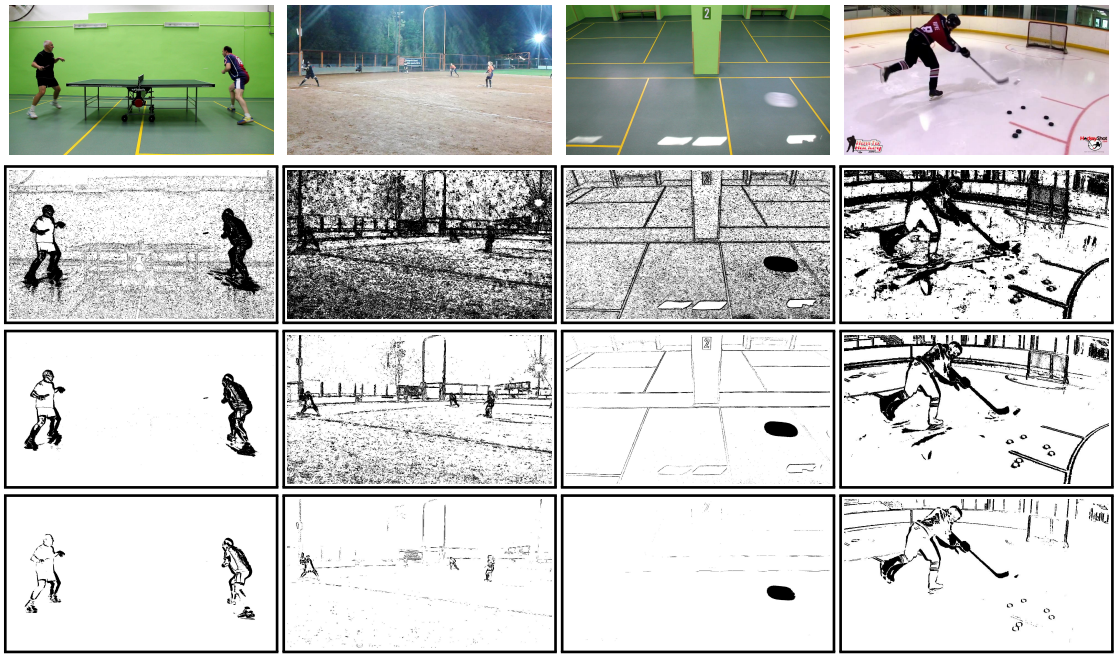

Figure 4.1. Selected frames from the FMOv2 dataset (top row) and binarised temporal difference images for thresholds (from second to bottom row) – 5, 25 and 50.

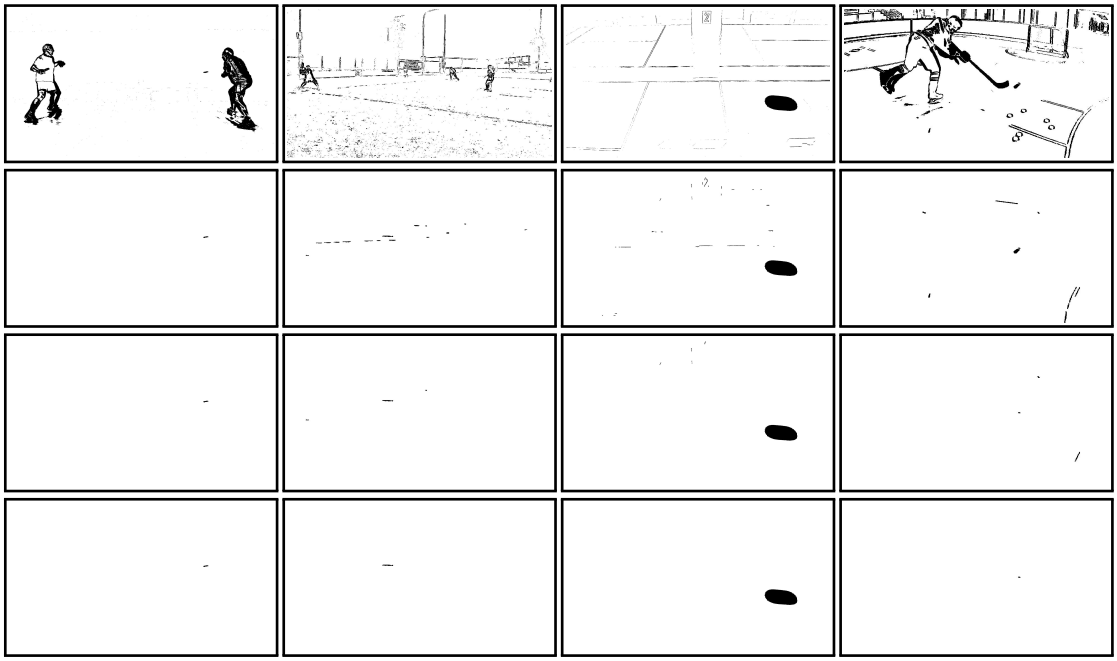

Figure 4.2. Connected component (CC) classification for binarised temporal difference images. Rows from top to bottom illustrate steps of the proposed FMO detector: 1. difference images for the thresholds estimated by Algorithm 2 (from left to right – 14, 31, 24, 26), 2. CC that are strokes, 3. strokes which are *not shadows nor illumination changes*, 4. strokes which are neither *lateral motion* – or FMOs.

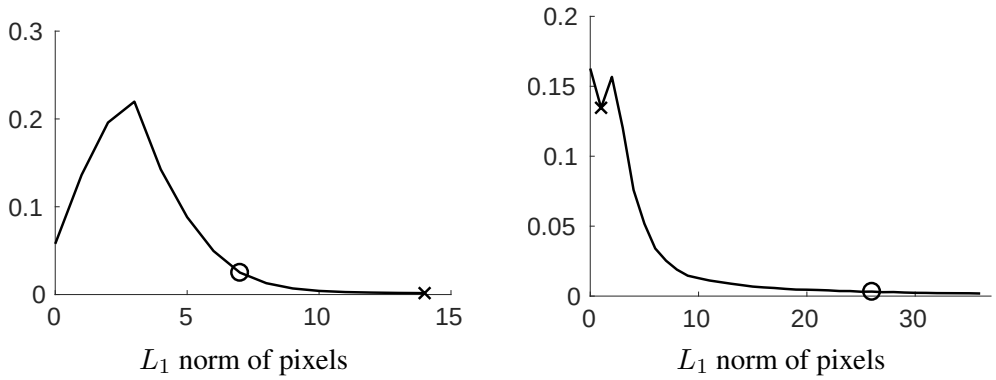

Figure 4.3. Normalised image histograms for temporal difference images. Circles denote the size threshold (see equation 4.1) which removes the majority of values and leaves a small part (10 %). Crosses indicate the derivative threshold (see equation 4.2) which sets a threshold, for which the binarised temporal difference images are stable to small changes. The estimated threshold is then the largest of 2 thresholds. The left histogram corresponds to the left image (tennis serve side) in Figure 4.1 with the estimated threshold 14 and the right histogram corresponds to the right image (hockey) in Figure 4.1 with the threshold 26.

Only small amount of values in Δ_t arise due to FMOs. Thus, we define the size threshold θ_s which leaves only $s \times 100\%$ highest values in Δ_t. Mathematically,

$$\theta_s = \max_{\theta} \theta \quad \text{subject to} \quad C(\theta) < 1 - s. \tag{4.1}$$

In evaluation section we set this threshold to 0.1 (10 %) based on the preliminary experiment that FMOs are almost always among 5 % of highest values in temporal differential images.

Another observation is that "the best" threshold should not be sensitive to small changes. In other words, we are looking for the value where the cumulative histogram is close to "flat" or equivalently where the histogram is close to zero. The derivative threshold θ_f is defined as the smallest value for which the derivative of the cumulative histogram is small. In the discrete case (values 0-255) it means

$$\theta_f = \min_{\theta} \theta \quad \text{subject to} \quad \frac{|C(\theta_f) - C(\theta_f + i)|}{C(\theta_f)} < \beta_f. \tag{4.2}$$

In the evaluation we set $i = 2$ and $\beta_f = 0.15$. The estimated threshold θ^* is then set to $\max(\theta_s, \theta_f)$, which guarantees that values of Δ_t at pixels affected by FMO are in a small part of values in Δ_t^b and small changes of the estimated threshold will not drastically change the difference image.

Examples of the threshold estimation for 2 images from Figure 4.1 are shown in Figure 4.3.

4.4. Stroke detection

According to the definition, FMOs are elongated in their direction of motion and form stroke-like regions. It can be seen as painting with a brush, where static FMO is a brush tail. In this section we use a method for measuring how connected components are close to stroke regions, which is similar to the text localisation method [25] – characters are also strokes.

4.4.1. Trajectory estimation

Every stroke is fully described by its trajectory and radius. We compute the distance transform of a binary image, which is enough to estimate them. In a simpler formulation, the distance

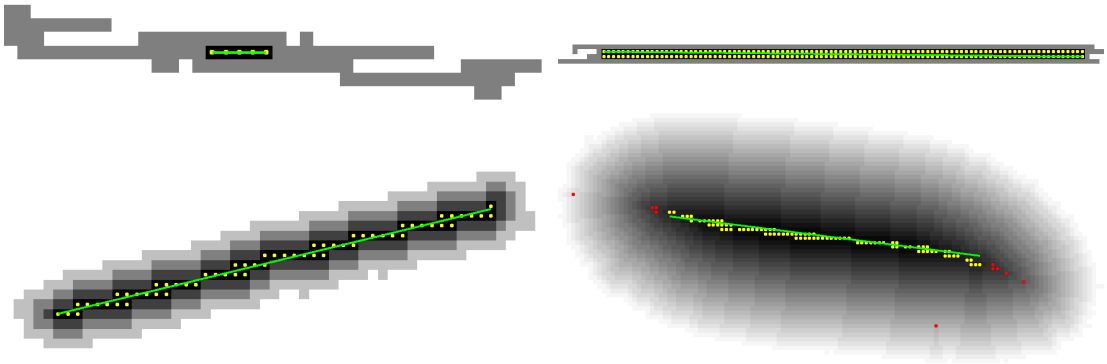

Figure 4.4. Trajectory estimation of regions and their strokeness. Pixel darkness denotes the distance transform (darker pixels mean larger values). Pixels with red/yellow dots indicate local maxima of the distance transform. Yellow pixels are trajectory pixels and green line is a fitted curve. Even though red pixels are local maxima, they are not considered trajectory (see Algorithm 2). The strokeness of regions from left-to-right and from top-to-bottom are 0.6641, 0.266, 0.0578, 0.0097.

transform is an operator which computes the Euclidean distance for each non-zero pixel to the closest zero pixel. The distance transform computation is a well-studied topic and can be done in linear time [27].

Every pixel on the trajectory of any stroke is a local maximum of the distance transform. Moreover, the value of the distance transform at any pixel on the trajectory is equal to the radius of that stroke. Thus, we compute the stroke radius as median of all its pixels which are local maxima. Next, pixels are considered to be part of the stroke trajectory if they are local maxima and their distance transform value is larger than a pre-defined fraction of the radius (β_r set to 0.8). The following algorithm 2 for trajectory estimation is based on these above-mentioned observations.

Algorithm 2 Trajectory estimator

1: $D = \text{dist}(\Delta_t^b)$ ▷ calculate the distance transform
2: $LM = \text{maxima}(D)$ ▷ find all local maxima
3: **for** CC $\in$ connectedComponents(Δ_t^b) **do** ▷ for every connected component in Δ_t
4: $\quad r_{cc} \leftarrow \text{median}(D[LM \cap CC])$ ▷ calculate radius
5: $\quad P_{cc} \leftarrow \{p \in LM \cap CC \mid D(p) > \beta_r r_{cc}\}$ ▷ find trajectory pixels
6: $\quad L_{cc} \leftarrow \text{fit}(P_{cc})$ ▷ fit a curve (e.g. straight line)
7: **end for**

The last step in the algorithm 2 estimates accurate trajectory by fitting a curve using LO-RANSAC (Locally Optimised RANdom SAmple and Consensus) [24]. When there is no contact, the curve can be restricted to a straight line. However, more advanced models (e.g. linear spline, parabolas) should be used when there is a contact (e.g. a ball bounces off the wall).

4.4.2. Strokeness

Strokeness is "the *stroke area ratio* feature which compares the actual area of a region with the ideal stroke area" [25]. Every connected component CC for which $abs(\frac{A_{cc}}{|CC|} - 1) < \beta_s$ (set to 0.4) is considered a stroke, where A_{cc} is the ideal stroke area and $|CC|$ is the real area of the connected component. To compute th ideal stroke area we use the weighted sum of distance transform values on the trajectory as in [25] with an extra term which adds semicircles on the

edges of the stroke:

$$A_{cc} = 2 \sum_{p \in P_{cc}} w_p D(p) + \pi r_{cc}^2, \qquad w_p = \frac{3}{\mathcal{N}_p}. \tag{4.3}$$

In equation (4.3) $\mathcal{N}_p$ denotes number of trajectory pixels within the 3×3 neighbourhood of pixel p. Weights w_p are used to normalise the stroke length when it has an even width (then there are two support pixel for one unit of stroke length) or when the trajectory has "alone" pixels (then some pixels are missing, and we have to compensate for it). Figure 4.4 includes examples of different regions and their strokeness estimation.

4.5. Taxonomy of CC in difference images

After image stabilisation, the PoC algorithm assumes that elongated connected components of the temporal difference image are caused by FMOs. This assumption leads to many false positives, such regions arise due to diverse phenomena: lateral motion of an object with long low-curvature edge, fast change of illumination on regions with stripe-like patterns, or on the edge of shadows. The phenomena are characterised by the properties of the gradient fields in the non-zero areas of the temporal difference image Δ_t and the corresponding area in images I_t and I_{t-1}. We propose an algorithm classifying connected components according to the cause of the non-zero response.

We learned a binary (FMO/Not-FMO) SVM classifier based on Histogram-of-Gradients (HoG) feature. The precision of the classifier was quite high (over 90%) considering the fact that it was learned only on the small amount of data. However, after we established all possible classes of connected components in Δ_t, it was shown that they can be classified based on simple tests on the gradient field.

4.5.1. Lateral motion of objects

Lateral motion of large objects or long and narrow objects can lead to connected components in Δ_t^b which have similar appearance as FMOs. However, if frame I_{t-1} is taken into consideration, they have totally different origin. Lateral motion is usually caused by small movements and thus traditional tracking algorithms can be applied. If a tracker succeeds to "track" a component, i.e. find it in a previous frame, and the displacement vector is lateral to the fitted curve of the component (as in Algorithm 2), then it is rejected.

The task of finding a connected component CC, given by a set of pixels, in a previous frame I_{t-1} can be described as a minimisation task:

$$h^* = \arg\min_h \|I_t(CC) - I_{t-1}(CC + h)\|. \tag{4.4}$$

The best displacement h^* can be found using the KLT tracking algorithm [28, 29], which is a simple iterative technique based on the image gradients. Other tracking techniques can be also used, but the main issue is speed of the algorithms. The test of being a "lateral motion" must be run a lot times for many connected components in every frame, thus it must be extremely fast. KLT tracker is simple and has this property – it is fast.

After the tracker has found the best displacement, the error should be normalised:

$$\text{err} = \frac{\|I_t(CC) - I_{t-1}(CC + h^*)\|}{\|I_t(CC) - I_{t-1}(CC)\|}. \tag{4.5}$$

Then, the displacement is considered as correct if it has $\text{err} < 0.5$. Figure 4.5 shows an example of how the lateral motion can look like and how the KLT tracker solves it.

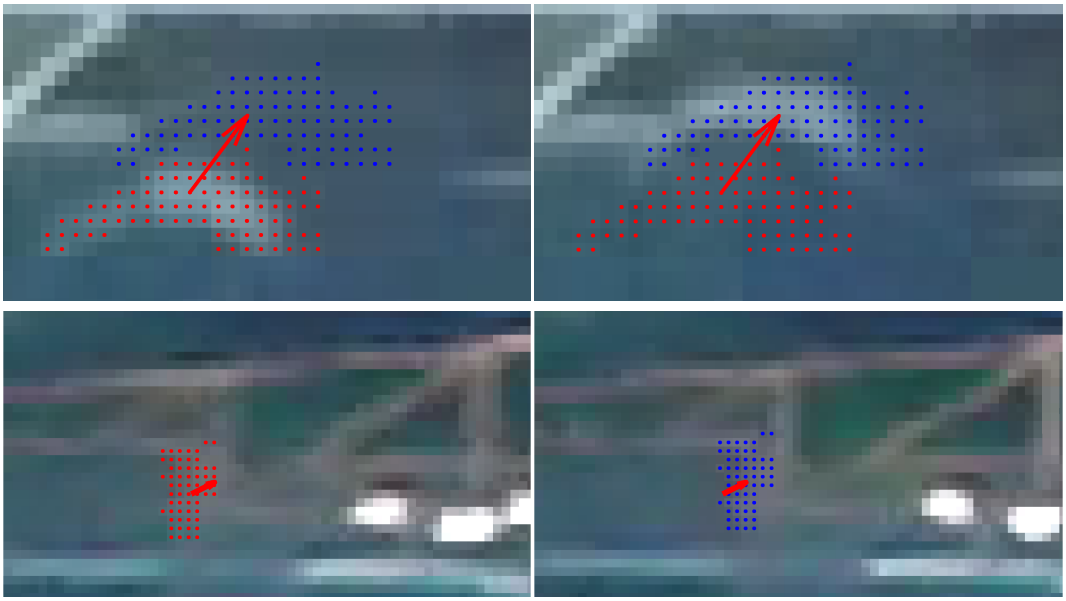

Figure 4.5. Displacements minimising the pre-defined loss (equation 4.4) are shown by red arrows. Red points – connected components in the current frame I_t, blue points – the displaced connected components in frame I_{t-1} which minimise the predefined loss. For better visualisation, images in the bottom row contain only a single component (either component in the current frame, or the displaced one).

4.5.2. Shadows and illumination changes

Another source of connected components in Δ_t are shadows and illumination changes. The gradient orientations and magnitudes do not change a lot in this regions comparing to the same regions in the previous frame. This is true only for CC which were caused by shadows or illumination changes. It makes them distinguishable from other CC, more importantly from FMOs.

Let G_t and O_t be gradient magnitude and orientation of the frame I_t. Then a connected component CC is classified as "shadows or illumination changes" if the following statements are true:

- mean $\|G_t(CC) - G_{t-1}(CC)\| < \beta_g$
- mean $\|O_t(CC) - O_{t-1}(CC)\| < \beta_o$

Some examples of components classified as part of this class are shown in Figure 4.6.

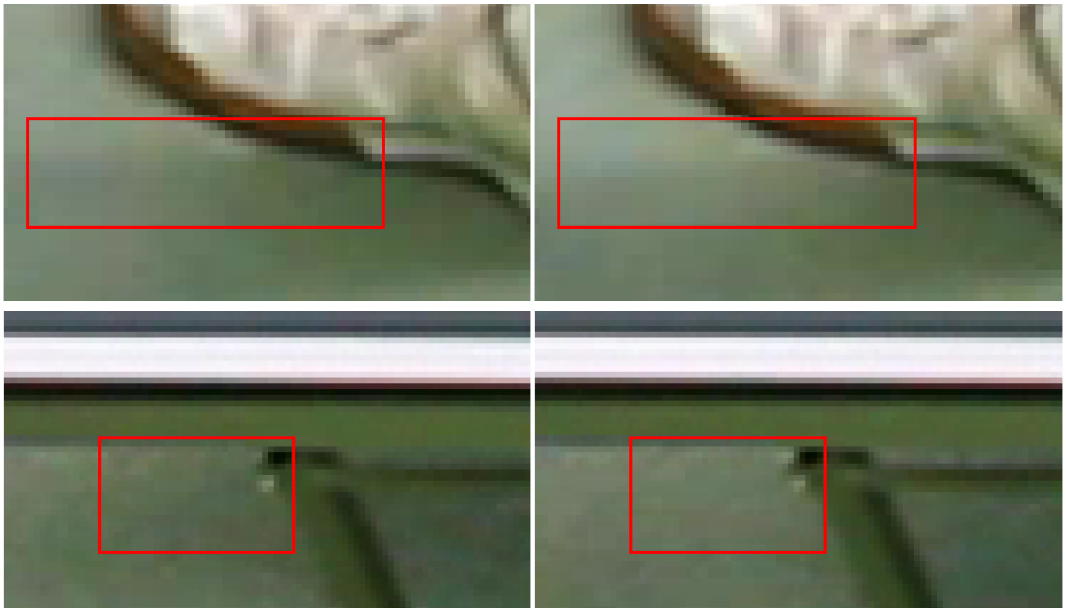

Figure 4.6. Examples of detected shadows and local illumination changes in images. Red bounding boxes show the detected regions.

5. Long-term FMO-SMO tracking

In many videos, objects are moving fast in some frames, then gradually or abruptly slow down or vice versa. The only available algorithm for FMO detection and tracking, the Proof-of-Concept algorithm [1], is "snake-like" – it fails on static and slow moving objects. In many applications, where complete analysis of the trajectory is required, this is unacceptable. The major advantage of continuous tracking of objects that are no longer in the FMO state is that it allows us to eventually establish the FMO appearance which is not corrupted by blurring. Figure 5.1 illustrates how integrated FMO-SMO tracking gives a full trajectory comparing to only detection of fast moving objects. Figure 5.3 shows an example how the speed may range for objects that were FMOs at least once – from 1 pixel length (static object, unblurred appearance) up to 105 pixels length (very fast FMO, severely blurred appearance). This section develops an algorithm which is able to handle the transition of fast to slow motion and vice versa, and integrates it with standard trackers that can handle the slow motion well.

5.1. Algorithm

We propose Algorithm 3 which combines FMO detection and continuous tracking of objects which have lost their FMO property – or slow moving objects (SMO). Additionally, this algorithm may include FMO tracking as proposed here [1]. However, if the FMO detector is accurate enough, this step is redundant. For better understanding how an object can change its states of being FMO or SMO and how the transition is handled, the reader is referred to Figure 5.2.

Quite often, videos contain multiple FMOs of the same or different object classes. Two objects are considered as of the same class if their unblurred full appearances are the same (e.g. "white table tennis ball" class, "yellow squash ball" class, etc). Thus, it is important to distinguish them and be able to learn the object appearance. We define $\mathcal{V}$ as the set of all known object classes, which are updated at every frame if some objects are localised. A single class representation will be discussed in section 5.3. All objects found at frame t are denoted by O_t.

After all SMOs are tracked, we remove them from the current frame by replacing with the background. This prohibits objects to be tracked and detected at the same time.

5.2. SMO tracking

When no information about the object is available a priori, we have to use some relative measurement for object speed. One option is to consider radii per exposure $[r/\tau]$. From the FMO definition follows that FMO speed must be more than $2\ r/\tau$. However, the proposed FMO detector can occasionally handle even slower movements, as well as standard trackers can sometimes handle some amount of blur. We observed that there is a performance overlap between FMO detector and standard trackers at the speed of near $2\ r/\tau$ which can be seen as a meeting point of FMO and SMO. Nevertheless, the deceleration (acceleration) of FMO can be large and the meeting point can be missed. Thus, an object is considered *slow* if its speed is less than $3\ r/\tau$ – and then the standard tracking is applied.

For each FMO which is considered *slow*, we initialise the CSR-DCF tracker [6] and use the output bounding box to create a temporal difference image at the region of interest. The

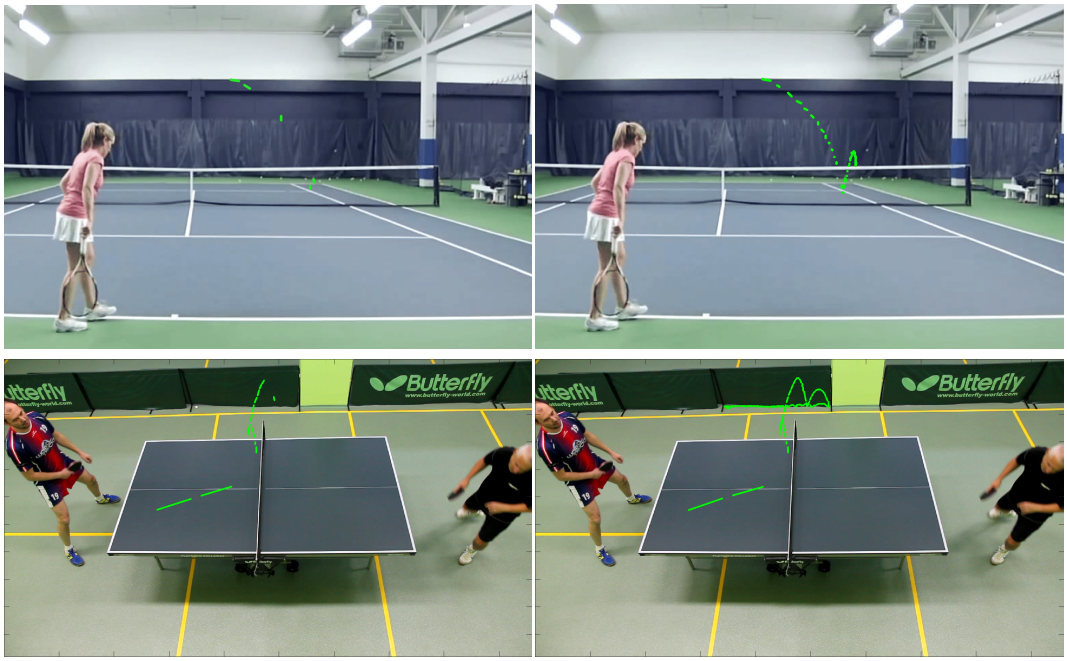

Figure 5.1. Examples of FMO and SMO tracking. Left images show FMO detections which missed many slowed down FMOs. Right images show FMO detections together with long-term tracking which give the full trajectory analysis. Green lines show object trajectories at multiple frames.

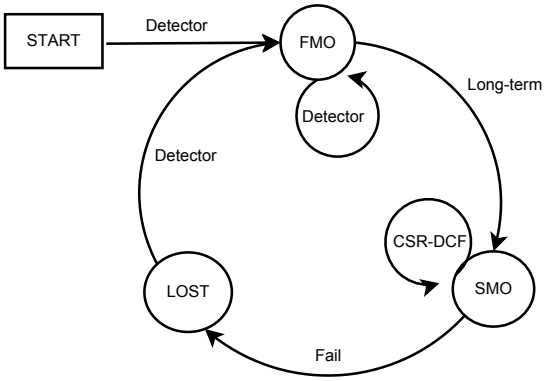

Figure 5.2. Long-term FMO-SMO tracking diagram.

SMO tracking is outlined in algorithm 4. Due to relatively small size of the bounding box, the algorithm is fast and gives either correct answer or a region with zero response in Δ. Two objects are considered *similar* if a newly found object corresponds to the class of the tracked object (discussed in section 5.3).

5.3. Object class assignment

Each FMO is assigned to some object class in a set of $\mathcal{V}$ and additionally to some instance in that class. For example, there can be 2 green balls and 3 red balls in one video – i.e. there are 2 object classes and they have 2 and 3 instances, respectively. An object class in represented by a set of blurred occurrences of its instances with different radii and lengths (length gives the amount of blur). However, in I_t we observe not only the blurred FMO but also some percentage of background as in equations (3.1) and (3.2). We subtract the background and store only the

Algorithm 3 Long-term FMO-SMO tracking

$t \leftarrow 0$
$\mathcal{V} \leftarrow \emptyset$ ▷ no known object classes
while hasNextFrame(video) **do**
    $I_t \leftarrow$ nextFrame(video)
    $O_t \leftarrow \emptyset$
    for $o \in O_{t-1}$ **do**
        if slow(o) **then**
            $O_t \leftarrow O_t \cap$ SMOtrack(I_{t-1}, I_t, o) ▷ see section 5.2
        end if
    end for
    remove(I_t, O_t) ▷ remove all already detected objects
    $O_t = O_t \cap$ detect(I_t) ▷ FMO detection as in chapter 4
    $\mathcal{V} \leftarrow$ update$(\mathcal{V}, O_t, O_{t-1})$ ▷ update obkect classes as in section 5.3
    $t \leftarrow t + 1$
end while

Algorithm 4 SMO tracking

tracker $\leftarrow$ init$(I_{t-1}, \text{boundingBox}(o))$ ▷ initialise tracker
bbx $\leftarrow$ track(tracker, I_t) ▷ find next location
$\Delta \leftarrow |I_t(\text{bbx}) - B_t(\text{bbx})| > \theta^*$ ▷ create difference image for the region of interest
CC* $\leftarrow \max(\text{CC} \in \Delta)$
if similar(CC*,o) **then**
    return CC*
else
    return $\emptyset$
end if

Figure 5.3. Object class representation of table tennis ball for 6 pixel radius. The top row show blurred appearances of the ball with different lengths (bottom row), which set the blurriness. The estimated percentage of the background was subtracted and only blurred object appearance is stored. The brightness of three longest blurred appearances was enhanced for better visualisation (producing some artefacts).

blurred appearances $\mathcal{H}_t F$:

$$\mathcal{H}_t F = I_t - (1 - \frac{1}{|P_t|}[P_t * F])B \tag{5.1}$$

Then, every object class stores a matrix of blurred appearances $\mathcal{H}_t F$ with row indices indicating radius and column indices indicating length. Rotation of every blurred appearance is normalised by making its fitted trajectory line parallel to the X axis. Figure 5.3 shows an example of a single row of such a matrix for an object class of a table tennis ball. Note, that both the static appearance, with a unit length, and a severely blurred appearances are observed and assigned to the same class. This makes several interesting applications possible, such as further, more precise estimations of the trajectory by de-blurring where the object if known.

Next step is to assign some object class to an unknown localised object. The similarity to some object class is defined as similarity to the closest stored appearance $\mathcal{H}_t F$ in that object class, or the blurred appearance which has the closest radius and length. Because some of the stored appearances can be shifted due to imprecise trajectory and radius estimation, we compute the normalised 2D cross-correlation [30] between the object and the closest appearance $\mathcal{H}_t F$. Then, the object is considered an inlier if the maximal value of cross-correlation is above some threshold β_i (set to 0.8). If no object class can be assigned to the object, a new class in $\mathcal{V}$ is created. At every update stage, we store all the blurred appearances to the object class.

6. Evaluation

6.1. FMOv2 Dataset

FMOv2 dataset[1] is an extended version of the FMO dataset [1] with more sequences, improved ground truth, in particular ground truth for slowed down FMOs and sequences with multiple objects. Moreover, it contains sequences with more challenging tasks, which do not involve sports, but without ground truth annotations.

The dataset contains videos of various activities involving fast moving objects, such as table tennis, tennis, frisbee, volleyball, squash, darts, arrows, softball, hockey. Besides sports activities, the dataset includes real-world situations which involve FMOs, such as falling fruits from trees (cherries, olives, apples), sparks from a circular saw, hailstorm rain, fireworks, fast toy cars (aka hot wheels) or flying insects.

Acquisition of the videos also differ: some are taken from a tripod with mostly static backgrounds, some have severe camera motions and dynamic backgrounds, some FMOs are nearly homogeneous, while some have coloured texture. Some of the videos were taken from YouTube

[1] Acquired together with Aleš Hrabalík as part of his Master thesis [31].

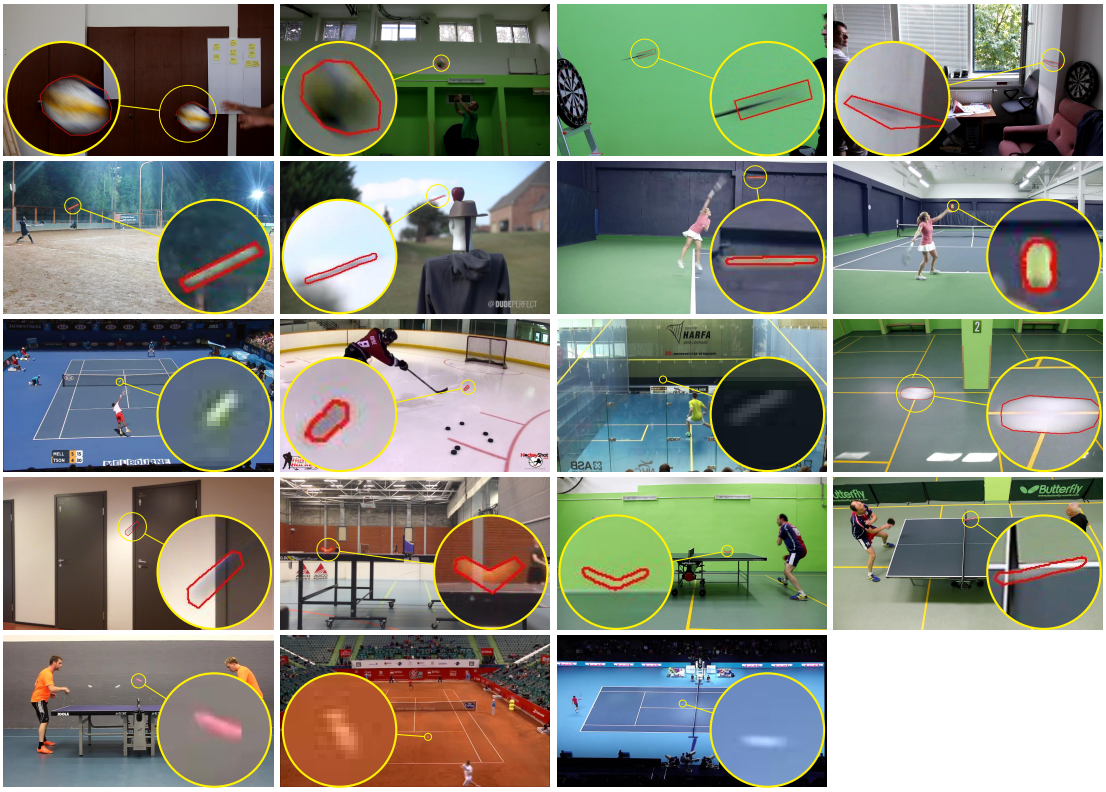

Figure 6.1. FMOv2 dataset – one example image per sequence. Comparing to FMO dataset, FMOv2 dataset has 3 more sequences (the bottom row). Red polygons delineate ground truth regions with fast moving objects. For clearer visualisation five frames do not show annotations because their area consists only of several pixels. The sequences are sorted from left to right and top to bottom as in Table 6.2.

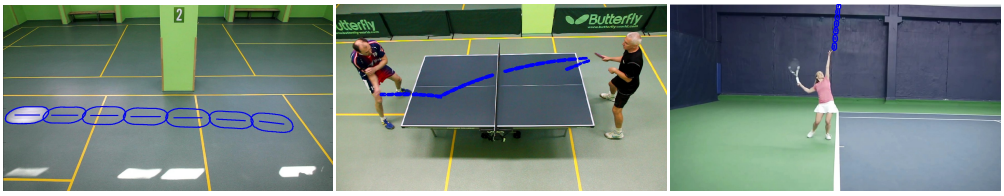

Figure 6.2. FMO detection and tracking. Each blue region represents object trajectory and contour in previous frames.

and all the links can be found in the Appendix. Most sequences are annotated with ground-truth locations of the object (even in cases when the object of interest does not strictly satisfy the notion of FMO).

An overview of the FMO dataset is in Figure 6.1, showing all included sequences which contain the ground-truth annotations. The FMOv2 dataset and ground-truth annotations are publicly available at `http://cmp.felk.cvut.cz/fmo/`.

6.2. Implementation

The proposed detector and long-term tracker were implemented in MATLAB. The code can be found in the attached CD. In this work, the speed of the methods was not the priority, nevertheless it is fast and can be implemented in real-time. Aleš Hrabalík in his Master thesis [31] focuses on real-time implementation of FMO detector in C++ and succeeds in this task.

6.3. Results

The proposed method was evaluated on both FMO and FMOv2 datasets. A true positive (TP) detection has an intersection over union (IoU) with the ground truth polygon greater than 0.5. All other detections are marked as false positives (FP). False negatives (FN) are FMOs in the ground truth with no associated detection. For example, if there is one detection and one ground-truth FMO and they do not intersect, both false positive count and false negative count are increased. The performance criteria are precision and recall:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}100\%, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}100\%. \tag{6.1}$$

There are two extreme cases which should be considered – when the denominators are equal to zero. If there are no detections at all (e.g. TP + FP = 0), the precision is defined as 100%. If there are no ground-truth FMOs (e.g. TP + FN = 0), the recall is defined as 100%. Another measure is the harmonic mean of precision and recall – F-score:

$$\text{F-score} = \frac{2}{\frac{1}{\text{Precision}}+\frac{1}{\text{Recall}}}100\% = \frac{2\text{TP}}{2\text{TP}+\text{FN}+\text{FP}}100\%. \tag{6.2}$$

Quantitative results for individual video sequences are listed in Table 6.1. All results were achieved for the same set of parameters as discussed in chapters 4 and 5. Both detector and long-term tracker are very precise – near 93%. It is a big improvement over the PoC algorithm [1], which is about 40% less precise. On the other hand, recall varies widely, ranging from 0% (no ground-truth FMOs were detected) for darts, volleyball and blue ball to 86% for the ping pong sequences. The sequences with the best results contain objects with prominent FMO characteristics, i.e. a large motion of almost spherical objects against a contrasting background.

			PoC alg. [1]		Detector		Long-term	
n	***Sequence name***	***#***	***Prec.***	***Recall***	***Prec.***	***Recall***	***Prec.***	***Recall***
1	volleyball	50	100.0	45.5	100.0	40.0	100.0	30.8
2	volleyball passing	66	21.8	10.4	100.0	0.0	100.0	0.0
3	darts	75	100.0	26.5	100.0	0.0	100.0	0.0
4	darts window	50	25.0	50.0	100.0	0.0	100.0	0.0
5	softball	96	66.7	15.4	84.6	39.3	84.6	39.3
6	archery	119	0.0	0.0	100.0	3.1	100.0	12.5
7	tennis serve side	68	100.0	58.8	91.0	55.6	92.9	72.2
8	tennis serve back	156	28.6	5.9	84.2	41.0	86.7	84.8
9	tennis court	128	0.0	0.0	95.8	41.1	83.9	50.0
10	hockey	350	100.0	16.1	100.0	7.7	100.0	1.6
11	squash	250	0.0	0.0	31.8	20.9	34.3	27.6
12	frisbee	100	100.0	100.0	100.0	75.0	100.0	80.0
13	blue ball	53	100.0	52.4	100.0	0.0	100.0	0.0
14	ping pong tampere	120	100.0	88.7	100.0	67.1	100.0	65.8
15	ping pong side	445	12.1	7.3	99.4	45.8	99.5	48.9
16	ping pong top	350	92.6	87.8	98.7	74.4	98.7	86.1
	Average	–	59.2	**35.5**	**92.9**	31.9	**92.5**	34.0

Table 6.1. Performance comparison between the proposed method and the Proof-of-Concept (PoC) algorithm [1]. Precision and recall on the FMO dataset is reported. For long-term FMO-SMO tracker the augmented ground-truth was used, i.e. with annotations for slowed down FMOs. Precision has increased by a wide margin with a minor loss in recall.

Next, we compare the results of the proposed detector to those of several standard state-of-the-art trackers, namely ASMS [15], DSST [11], SRDCF [12], MEEM [16], and STRUCK [17]. The results are presented in Table 6.3 in terms of the percentage of frames with a successful detection, which is equivalent to recall of the detector. Some of the standard trackers performed reasonably well on sequences, where the motions are relatively slow (e.g. volleyball, frisbee), but overall results are poor. The proposed method performs significantly better. This is understandable because the compared methods were not designed for scenarios involving FMOs, but it highlights the need for a specialised FMO detector and tracker.

Videos with detection results are included in the attached CD or can be found online at `http://cmp.felk.cvut.cz/fmo/demo/`. Several examples are shown in Figure 6.2, where the reader can see the detected trajectory and boundary at consequent frames.

6.4. Limitations and failure cases

In this section we discuss limitations of the proposed method and other failure cases. False negatives, or not localised FMOs, occur in these types of situations:

- The object motion is too small (e.g. archery, volleyball, hockey) and it cannot be initially detected, because the FMO definition is not satisfied. This happens because the FMOv2 dataset contains ground-truth annotations of objects which do not always satisfy the notion of FMO. Thus, the dataset is too challenging for the proposed method. We consider this fact as positive and the one which motivates researchers to work on this problem.
- The object itself is considerably different from a sphere (e.g. darts, archery). This is a limitation of the proposed method because it assumes a spherical object. However, a method which could detect and track any type of FMOs will be likely based on deconvolution which

	Sequence name	#	TP	TN	FP	FN	Prec.	Recall	F-score
1	volleyball	50	4	37	0	9	100.0	30.8	47.1
2	volleyball passing	66	0	0	0	66	100.0	0.0	0.0
3	darts	75	0	39	0	36	100.0	0.0	0.0
4	darts window	50	0	45	0	5	100.0	0.0	0.0
5	softball	96	11	68	2	17	84.6	39.3	53.7
6	archery	119	4	87	0	28	100.0	12.5	22.2
7	tennis serve side	68	13	50	1	5	92.9	72.2	81.3
8	tennis serve back	156	26	97	4	33	86.7	44.1	58.4
9	tennis court	128	26	61	5	38	83.9	40.6	54.7
10	hockey	350	1	293	0	61	100.0	1.6	3.2
11	squash	250	36	140	69	98	34.3	26.9	30.1
12	frisbee	100	16	80	0	4	100.0	80.0	88.9
13	blue ball	53	0	32	0	21	100.0	0.0	0.0
14	ping pong tampere	120	46	44	0	30	100.0	60.5	75.4
15	ping pong side	445	219	29	1	217	99.5	50.2	66.8
16	ping pong top	350	385	16	5	67	98.7	85.2	91.4
17	more balls	300	1013	13	61	227	94.3	78.5	85.7
18	tennis court 2	278	72	59	6	147	92.3	32.8	48.5
19	atp serve	655	313	192	111	150	73.8	67.6	70.6
	Average	–	115	72.7	13.9	68.9	91.6	38.1	46.2

Table 6.2. Performance of the long-term FMO-SMO tracker on FMOv2 dataset which extends FMO dataset. We report number of True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), Precision, Recall and the F-score. Number of frames is shown in the column with #.

is a difficult task so far because of the computational complexity.

- The background is too similar to the object colour (e.g. table tennis net, white edge of the table), obstacles or other moving objects near FMO (e.g. hand, tennis rocket). It is another failure case of the proposed method.

False positives sometimes appear when local movements of larger objects, such as stripes on clothes or dots on rockets, which move in the direction of their longer axis, can be partially explained by the FMO model, or due to imprecise camera stabilisation.

6.5. Applications

6.5.1. FMO localisation

The direct output of the proposed method is the position of localised fast moving objects, e.g. all pixels affected by the FMO, trajectory, radius, etc. Besides many obvious applications important for sports analytics, FMO localisation can be used for real-world situations. We observed that FMOs appear during hailstorm rains, fireworks or other types of explosions, sparks (e.g. from a circular saw), or any other falling or shooting objects. Figure 6.4 contains examples of localised fast moving objects in the above-mentioned situations. Note, that it is not important to detect all present FMOs. In such situations, usually only FMO presence is of interest.

FMO localisation can also operate as a detector of explosions (middle and bottom rows in Figure 6.4). Imagine a camera in front of a dangerously explosive element which can immediately observe the explosion and turn on the defence system or alarm. Automotive vehicles can make use of FMO localisation to detect flying objects in front of them (e.g. stones, insects).

Sequence name	ASMS	DSST	MEEM	SRDCF	STRUCK	Detector
volleyball	**80**	0	50	0	10	40
volleyball passing	12	6	**95**	88	8	0
darts	3	0	**6**	0	0	0
darts window	0	0	0	0	0	0
softball	0	0	0	0	0	**39**
archery	**5**	**5**	**5**	**5**	0	3
tennis serve side	7	0	0	0	6	**56**
tennis serve back	5	0	0	0	3	**41**
tennis court	0	0	3	3	0	**41**
hockey	0	0	0	0	0	**7**
squash	0	0	0	0	0	**21**
frisbee	65	0	6	6	0	**75**
blue ball	**30**	0	0	0	25	0
ping pong tampere	0	0	0	0	0	**67**
ping pong side	1	0	0	0	0	**46**
ping pong top	0	0	0	0	1	**74**
Average	17	1	1	1	3	**32**

Table 6.3. Performance of baseline methods (ASMS[15], DSST[11], MEEM[16], SRDCF[12], STRUCK[17]) on FMO dataset. We report percentage of presented FMOs where tracking was successful (IoU > 0.5), which is equivalent to recall of the detector.

6.5.2. Temporal Super-Resolution

Another possible application of the proposed method is the task of temporal super-resolution, which increases the video frame-rate by filling out the gap between existing frames and artificially decreases the exposure period of existing frames. Let define the temporal n-fold super-resolution as an operation which replaces every frame I_t by a set of frames $\{I_t^0, \ldots, I_t^{n-1}\}$.

The naive approach for this task is the plain interpolation of adjacent frames, which is inadequate for videos containing FMOs. This operation is equivalent to the convex combination of adjacent frames:

$$I_t^i = (1 - \frac{i}{n})I_t + \frac{i}{n}I_{t+1} \tag{6.3}$$

This produces natural slow-motion for *slow* objects, but makes FMOs longer and more transparent, which has a negative visual effect (see Figure 6.5).

A more precise approach requires moving objects to be localised, de-blurred (by deconvolution [1]), and their motions modelled, which the proposed method accomplishes, so that new frames can be synthesised at the desired frame-rate. Any frame-rate can be achieved using the FMO formation model in equation (3.1). All other parts of image frames, which do not contain FMOs, can be synthesised using the plain interpolation. Figures 6.3, 6.5 show example results of the temporal super-resolution and compare the proposed method with plain interpolation.

In Figure 6.6 we illustrate the result of FMO de-blurring in the form of temporal super-resolution. The top image shows a frame captured by a conventional video camera (25fps), which contains a volleyball that is severely motion blurred. Frames in red boxes show frames captured by a high-speed video camera (250fps) spanning approximately the same time frame – the volleyball flies from left to right while rotating clockwise. Frames in blue boxes show the result of FMO de-blurring, computed solely from the single frame (the top image), at times corresponding to the high-speed frames above. The restoration is on par with the high-speed ground-truth; it significantly enhances the video information content merely by post-

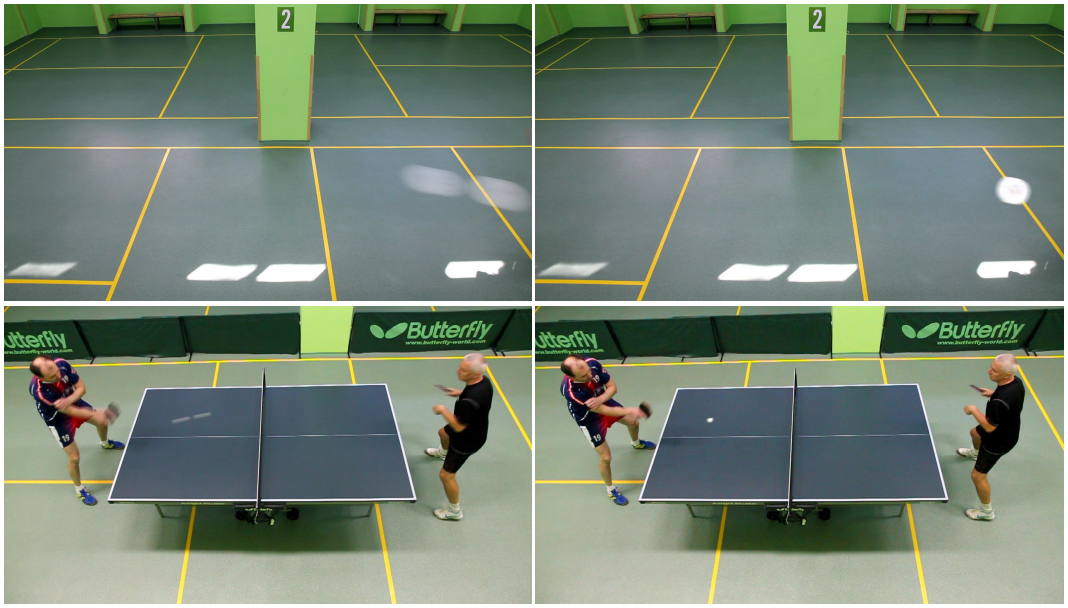

Figure 6.3. Temporal super-resolution using plain interpolation (left images) and the appearance estimation model (right images). Videos with temporal super-resolutions can be found online at `http://cmp.felk.cvut.cz/fmo/demo/super-resolution/` or in the attached CD. Two sequences shown here are frisbee_tsr.avi and ping_pong_top_tsr.avi.

processing. For comparison, we also display the calculated rotation axis and the one estimated from the high-speed video. Both are close to each other; compare the blue cross and red circle in the Figure 6.6. Note that for a human observer it is impossible to determine the ball rotation from blurred images while the proposed algorithm with the temporal super-resolution output provides this insight. Another appearance estimation example is in Figure 6.3, where we use the simplified model of pure translation motion.

6.5.3. FMO highlighting

Another popular use case is highlighting FMO in sport videos. Due to the extreme blur and small size, FMOs are often hard to localise, even for humans, despite having the context provided by perfect semantic scene understanding. Simple highlighting, like recolouring or scaling, enhances the viewer's experience. The bottom row in Figure 6.5 demonstrates FMO highlighting by rescaling, recolouring or increasing the exposure fraction.

6.5.4. FMO removal

Next logical application is FMO removal. It may be useful especially for clearing videos from unwanted FMOs, such as hailstorm or other flying objects (the top row in Figure 6.4). If all the parameters of an FMO would be perfectly estimated, and its speed is faster than 2 radii per exposure (i.e. the background is partially visible), it would be possible to remove the blurred appearance of the FMO, which can be calculated by deconvolution. But in general the estimated properties of an FMO are not perfect, and it cannot be entirely removed. Then, another way to remove all FMOs from a video frame is replacing it with the background. The bottom row in Figure 6.5 also illustrates FMO removal. Video with illustration of FMO removal can be found either online or in the attached CD as ping_pong_top_remove.avi.

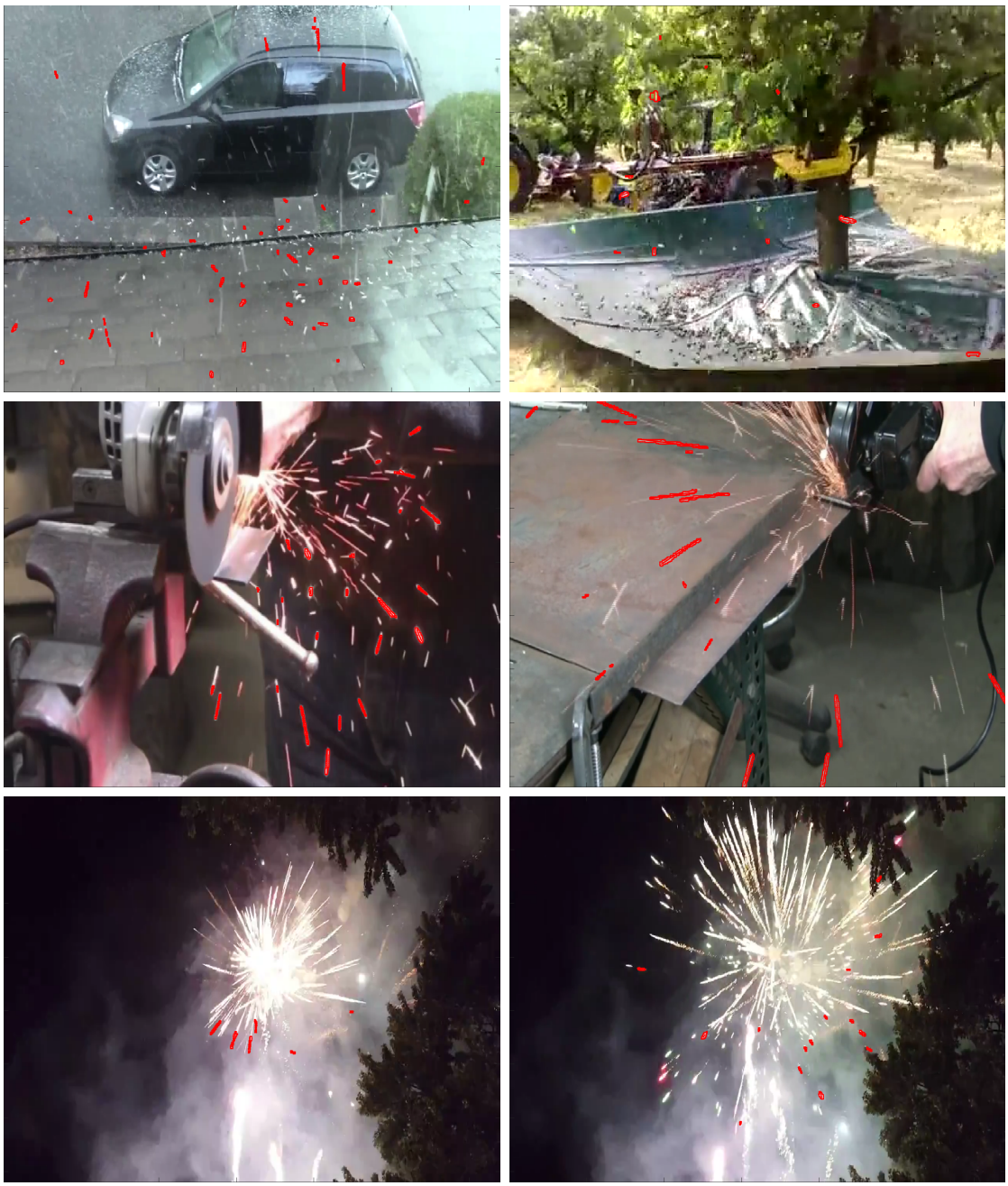

Figure 6.4. FMO localisation by the proposed method on non-sports videos. All localised FMOs in the current frame are marked by red colour. Videos with FMO localisation can be found online at `http://cmp.felk.cvut.cz/fmo/demo/localisation/` or in the attached CD. Images contain from left to right and from top to bottom: 1. hailstorm rain, 2. falling apples and leafs, 3. sparks from a circular saw, 4. sparks from a circular saw, 5-6. fireworks. The displayed frames comprise up to 50 localised fast moving objects. The bottom row with fireworks does not contain many detections because the FMO formation model is not satisfied for fireworks – they additionally emit light and change their shape.

Figure 6.5. Top and middle rows: comparison of temporal super-resolution by 8x using plain interpolation (top row) and using interpolation with FMOs (middle row). Bottom row: example of different application, from left to right – original FMO, removed FMO, replaced by the FMO formation model (i.e. expected perfect FMO), FMO with full exposure time (exposure fraction equals 1), increased radius, decreased radius, changed colour, changed colour and increased radius.

6.5.5. Speed estimation

Object speed is already estimated in radii per exposure during the detection step. If the camera frame rate is known in advance (which is commonly a case), it is easy to convert speed to radii per second. Then, such a simple information as the real radius of FMO will make a speed conversion to real world units (e.g. km/h, mph) possible. Speed estimation has many applications for sports analysis. For instance, many professional tennis players are interested in their serve speed. FMO detection and tracking can replace radar guns, which are used to measure the serve speed nowadays.

6.5.6. Exposure time and fraction estimation

Blur in consecutive video frames provides important information about the sensor, such as exposure fraction or shutter type. This gives many hints for sensor identification. The exposure fraction can be easily calculated if the same FMO has been localised in consecutive frames.

6.5.7. Other applications

The proposed method can also find its applications in different areas, e.g. in mechanics – blur gives us information about fast vibrating objects; in scanning probe microscopy – blur encodes the measuring tip shape and scanning procedure; in ophthalmology - estimated blur in retinal images caries information about pupil abnormalities.

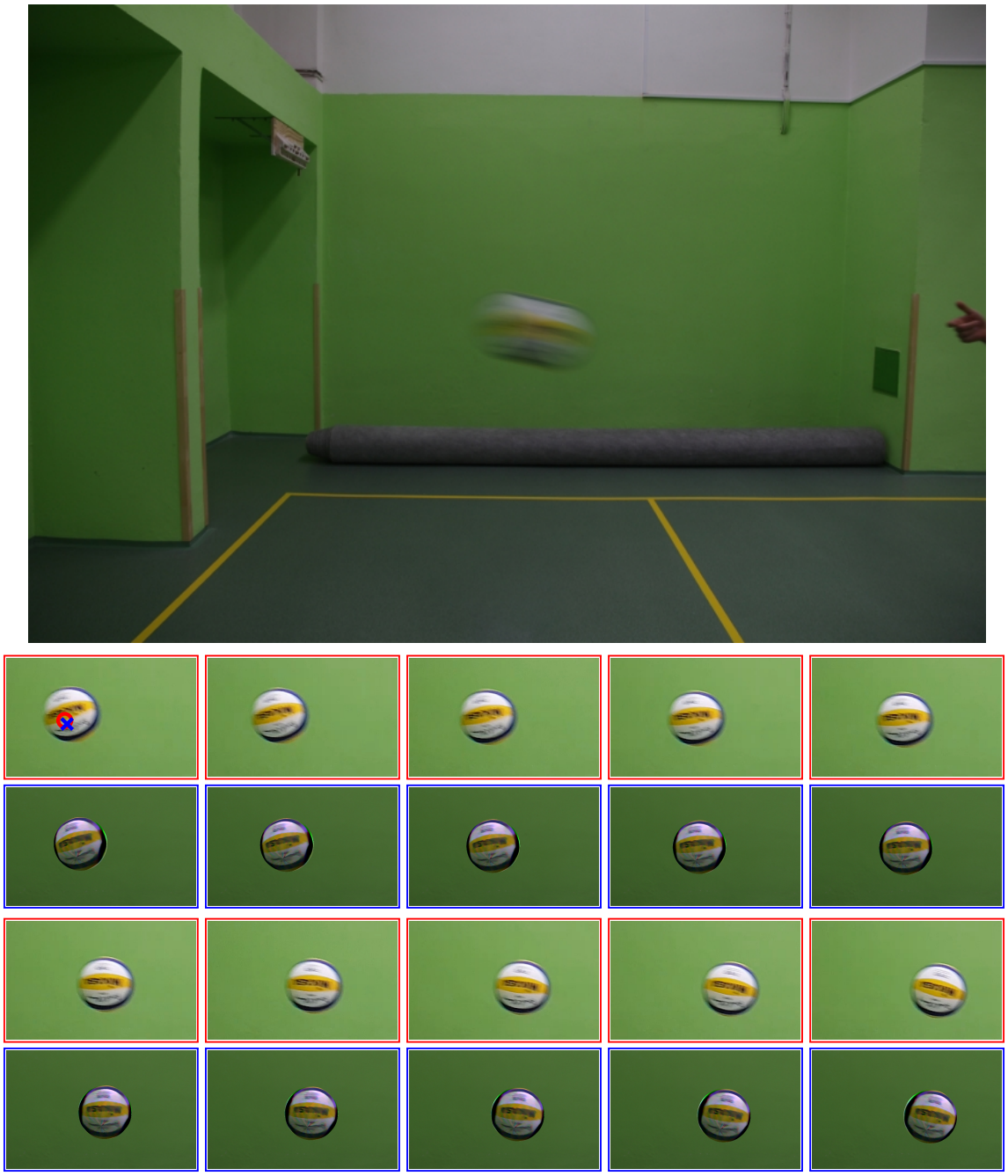

Figure 6.6. Reconstruction of a volleyball blurred by motion and rotation. Top image: input video frame. Frames in red boxes: actual frames from a high-speed camera (250fps). Frames in blue boxes: frames at corresponding times reconstructed from a single frame of a regular camera (25fps), i.e. 10x temporal super-resolution. The first frame from a high-speed camera shows the rotation axis position estimated from the blurred frame (blue cross) and from the highspeed video (red circle). The video can be found online or in the attached CD as volleyball_tsr_1frame.avi. Courtesy of Jan Kotera and Filip Šroubek.

7. Conclusions

We covered the problem of continuous tracking of objects which can be fast moving (FMO). The FMO detector has been proposed which can discover previously unseen object but only as FMOs. It is robust and generic method, i.e. not requiring prior knowledge of appearance. The FMO detector is part of the proposed long-term FMO-SMO tracker, which combines it with the state-of-the-art tracker CSR-DCF [6]. It was shown that in many cases object that were fast moving can abruptly slow down or vice versa. Experimentally we observed that the proposed integrated algorithm can successfully handle this transition.

FMOv2 dataset consisting of 19 sports videos with ground-truth annotations is introduced, which extends the FMO dataset [1]. In addition to sports activities with annotations, the dataset includes real-world situations which involve FMOs, such as falling fruits from trees (cherries, olives, apples), sparks from a circular saw, hailstorm rain, fireworks, hot wheels toy cars or flying insects.

Tracking FMOs is considerably different from standard object tracking targeted by state-of-the-art algorithms and thus requires a specialised approach. The proposed method outperforms baseline methods by a wide margin on FMO and FMOv2 datasets. One of the main advantages is that the proposed long-term tracker is very precise in discovering and tracking objects that alter their state of being fast moving and slow moving. Among the discusses applications, the most important are applications in sports analytics, such as realistic increase of video frame-rate (temporal super-resolution), artificial object highlighting, visualisation of rotational axis and measurement of speed and angular velocity.

Bibliography

[1] D. Rozumnyi, J. Kotera, F. Sroubek, L. Novotny, and J. Matas, "The world of fast moving objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 10, 17, 21, 22, 23, 25, 31

[2] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 261–271, Feb. 2007. 1, 3

[3] B. Babenko, M. H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1619–1632, Aug 2011. 1, 3

[4] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Comput. Vis. Image Underst.*, vol. 117, pp. 1245–1256, Oct. 2013. 1, 3

[5] M. Kristan *et al.*, *The Visual Object Tracking VOT2016 Challenge Results*, pp. 777–823. Cham: Springer International Publishing, 2016. 1, 3

[6] A. Lukezic, T. Vojir, L. Cehovin, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 17, 31

[7] M. A. Zaveri, S. N. Merchant, and U. B. Desai, "Small and fast moving object detection and tracking in sports video sequences," in *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, vol. 3, pp. 1539–1542 Vol.3, June 2004. 3

[8] A. V. Kruglov and V. N. Kruglov, "Tracking of fast moving objects in real time," *Pattern Recognition and Image Analysis*, vol. 26, no. 3, pp. 582–586, 2016. 3

[9] H. Jin, P. Favaro, and R. Cipolla, "Visual tracking in the presence of motion blur," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 18–25 vol. 2, June 2005. 3

[10] T. A. Biresaw, A. Cavallaro, and C. S. Regazzoni, "Correlation-based self-correcting tracking," *Neurocomput.*, vol. 152, pp. 345–358, Mar. 2015. 3

[11] M. Danelljan, G. Hauml;ger, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, BMVA Press, 2014. 3, 23, 25

[12] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4310–4318, 2015. 3, 23, 25

[13] C. Tomasi and T. Kanade, *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991. 3

[14] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 564–575, May 2003. 3

[15] T. Vojir, J. Noskova, and J. Matas, *Robust Scale-Adaptive Mean-Shift for Tracking*, pp. 652–663. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. 3, 23, 25

[16] J. Zhang, S. Ma, and S. Sclaroff, *MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization*, pp. 188–203. Cham: Springer International Publishing, 2014. 3, 23, 25

[17] 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*, vol. 38, pp. 2096–2109, Oct 2016. 3, 23, 25

[18] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 1442–1468, July 2014. 3

[19] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking vot2015 challenge results," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015. 3

[20] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3

[21] B. Tordoff and D. W. Murray, *Guided Sampling and Consensus for Motion Estimation*, pp. 82–96. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. 8

[22] E. Rosten and T. Drummond, *Machine Learning for High-Speed Corner Detection*, pp. 430–443. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. 8

[23] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 510–517, June 2012. 8

[24] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Joint Pattern Recognition Symposium*, pp. 236–243, Springer Berlin Heidelberg, 2003. 8, 13

[25] L. Neumann and J. Matas, "Efficient scene text localization and recognition with local character refinement," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, (California, US), pp. 746–750, IEEE, Aug 2015. 9, 12, 13

[26] A. Fabijanska and D. Sankowski, "Image noise removal - the new approach," in *2007 9th International Conference - The Experience of Designing and Applications of CAD Systems in Microelectronics*, pp. 457–459, Feb 2007. 10

[27] C. R. Maurer, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 265–270, Feb 2003. 13

[28] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," 1981. 14

[29] C. Tomasi and T. Kanade, "Detection and tracking of point features," 1991. 14

[30] J. Lewis, "Fast normalized cross-correlation," in *Vision interface*, vol. 10, pp. 120–123, 1995. 20

[31] A. Hrabalík, "Implementing and applying fast moving object detection on mobile devices," in *Master Thesis*, 2017. 21, 22

A. Video sources

Sequence name	Origin / Recorded by
volleyball	Filip Šroubek team
volleyball passing	Filip Šroubek team
darts	Filip Šroubek team
darts window	Filip Šroubek team
softball	Jana Nosková
archery	https://youtu.be/eCtb_ylVDvU
tennis serve side	Filip Šroubek team
tennis serve back	Filip Šroubek team
tennis court	https://youtu.be/uy1ULXjkM-E
hockey	https://youtu.be/lsxYCuu-DUY
squash	https://youtu.be/OcYC4bjElZs
frisbee	Filip Šroubek team
blue ball	Denys Rozumnyi
ping pong tampere	Denys Rozumnyi
ping pong side	Filip Šroubek team
ping pong top	Filip Šroubek team
tennis court 2	https://youtu.be/uy1ULXjkM-E
hailstorm	https://youtu.be/l748t-r7VmQ
falling cherries	https://youtu.be/ykGuOIMGbLI
falling apples	https://youtu.be/kKCHEFxNkmM
falling olives	https://youtu.be/HxHOkQ1VilM
falling walnuts	https://youtu.be/uFcDzjHxM5E
hot wheels toy cars	https://youtu.be/YfWU_uIZBtc
sparks	https://youtu.be/jTlzcDJqcWU
ping pong 5 balls	https://youtu.be/Y1EOHFhmHYs

Table A.1. The table with origins of video sequences. The bottom part of the table contains videos without ground-truth annotations.

B. CD content

- `/`
  - `thesis` ..... LaTeX source code for the thesis, including thesis.pdf
  - `dataset` ..... FMOv2 dataset
    - `seq`
      - `with_gt` ... Sports sequences, for which ground truth annotations are available
      - `no_gt` ..... Various real-world sequences without ground truth annotations
    - `qt` ..... The ground truth annotations
  - `src` ..... MATLAB source code used for the evaluation
    - `go.m` ..... The main file for testing, usage go(<url>)
    - `...` ..... Other source files, dependency on CSR-DCF (contact Tomáš Vojíř)
  - `demo`
    - `localisation` ..... Examples of FMO localisation
    - `super-resolution` ..... Examples of temporal super-resolution
    - `removal` ..... Examples of FMO removal

C. Used parameters

Parameter	Value
β_s	0.4
β_f	0.15
β_g	0.025
β_o	0.2
β_i	0.8
β_r	0.8
i	2
s	10 %

Table C.1. Values of parameters used for the evaluation.