The World of Fast Moving Objects

Denys Rozumnyi[1,3] Jan Kotera[2] Filip Šroubek[2] Lukáš Novotný[1] Jiří Matas[1]
[1]Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University in Prague
[2]Institute of Information Theory and Automation, Czech Academy of Sciences
[3]Department of Signal Processing, Tampere University of Technology

Abstract

The notion of a Fast Moving Object (FMO), i.e. an object that moves over a distance exceeding its size within the exposure time, is introduced. FMOs may, and typically do, rotate with high angular speed. FMOs are very common in sports videos, but are not rare elsewhere. In a single frame, such objects are often barely visible and appear as semi-transparent streaks.

A method for the detection and tracking of FMOs is proposed. The method consists of three distinct algorithms, which form an efficient localization pipeline that operates successfully in a broad range of conditions. We show that it is possible to recover the appearance of the object and its axis of rotation, despite its blurred appearance. The proposed method is evaluated on a new annotated dataset. The results show that existing trackers are inadequate for the problem of FMO localization and a new approach is required. Two applications of localization, temporal super-resolution and highlighting, are presented.

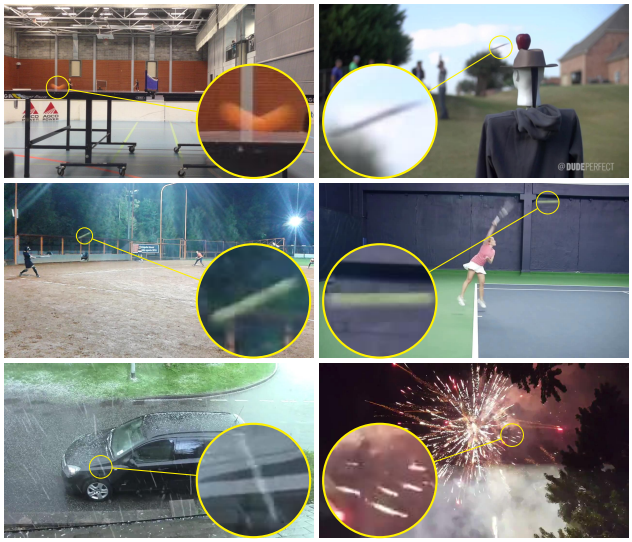


Figure 1: Fast moving objects appear as semi-transparent streaks larger than their size. Examples (left-to-right, top-to-bottom) from table tennis, archery, softball, tennis, hailstorm and fireworks.

1. Introduction

Object tracking has received enormous attention by the computer vision community. Methods based on various principles have been proposed and several surveys have been compiled [2, 3, 11]. Standard benchmarks, some comprising hundreds of videos, such as ALOV [22], VOT [15, 16] and OTB [27] are available. Yet none of them include objects that are moving so fast that they appear as streaks much larger than their size. This is a surprising omission considering the fact that such objects commonly appear in diverse real-world situations, in which sports play undoubtedly a prominent role; see examples in Fig. 1

To develop algorithms for detection and tracking of fast moving objects, we had to collect and annotate a new dataset. The substantial difference of the FMO dataset and the standard ones was confirmed by ex-post analysis of inter-frame motion statistics. The most common overlap of ground truth bounding boxes in two consecutive frames is *zero* for the FMO set while it is *close to one* for ALOV, OTB and VOT [22, 27, 15]. The speed of the tracked object projected to image coordinates, measured as the distance of object centers in two consecutive frames, is on average ten times higher in the new dataset, see Fig. 2. Given the difference in the properties of the sequences, it is not surprising that state-of-the-art trackers designed for the classical problem do not perform well on the FMO dataset. The two "worlds" are so different that on almost all sequences the classical state-of-the-art methods fail completely, their output bounding boxes achieving a 50% overlap with the ground truth in *zero* frames, see Tab. 3.

In the paper, we propose an efficient method for FMO localization and a method for estimation of their extrinsic – the trajectory and the axis and angular velocity of rotation, and intrinsic properties – the size and color of the object. In specific cases we can go further and estimate the full ap-

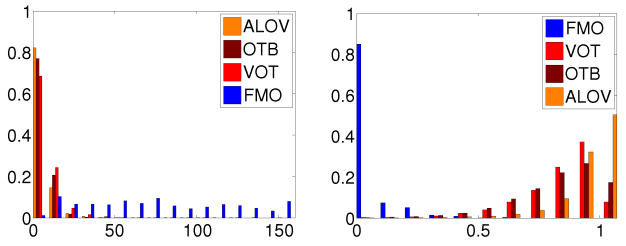


Figure 2: The FMO dataset includes motions that are an order of magnitude faster than three standard datasets - ALOV, VOT, OTB [22, 15, 27]. Normalized histograms of projected object speeds (left) and intersection over union IoU of bounding boxes (right) between adjacent frames.

pearance model of the object. Properties like the rotation axis, angular velocity and object appearance require precise modeling of the image formation (acquisition) process. The proposed method thus proceeds by solving a blind space-variant deconvolution problem with occlusion.

Detection, tracking and appearance reconstruction of FMOs allow performing tasks with applications in diverse areas. We show, for instance, the ability to synthesize realistic videos with higher frame rates, i.e. to perform temporal super-resolution. The extracted properties of the FMO, such as trajectory, rotation angle and velocity have application, *e.g.* in sports analytics.

The rest of the paper is organized as follows: Related work is discussed in Section 2. Section 3 defines the main concepts arising in the problem. Section 4 explains in detail the proposed method for FMO localization. The estimation of intrinsic and extrinsic properties formulated as an optimization problem is presented in Section 5. In Section 6, the FMO annotated dataset of 16 videos is introduced. Last, the method is evaluated and its different applications are demonstrated in Section 7.

2. Related Work

Tracking is a key problem in video processing. A range of methods has been proposed based on diverse principles, such as correlation [4, 9, 8], feature point tracking [24], mean-shift [6, 25], and tracking-by-detection [29, 12]. The literature sometimes refer to fast moving objects, but only the case with no significant blur is considered, *e.g.* [28, 17].

Object blur is a cue for object motion, since the blur size and shape encode information about motion. However, classical tracking methods suffer from blur, yet FMOs consist predominantly of blur. Most motion deblurring methods assume that the degradation can be modeled locally by a linear motion. One category of methods works with occlusion and considers the object's blurred transparency map [13]. Blind deconvolution of the transparency map is easier, since the latent sharp map is a binary image. The same idea applied to rotating objects was proposed in [21]. An interesting variation was proposed in [7], where linear motion blur is estimated locally using a relation similar to optical flow. The main drawback of these methods is that an accurate estimation of the transparency map using alpha matting algorithms [18] is necessary.

Methods exploiting the fact that autocorrelation increases in the direction of blur were proposed to deal with objects moving over static backgrounds [5, 14]. Similarly [19, 23], autocorrelation was considered for motion detection of the whole scene due to camera motion. However, all these methods require a relatively large neighborhood to estimate blur parameters, which means that they are not suitable for small moving objects. Simultaneously dealing with rotation of objects has not been considered in the literature so far.

3. Problem definition

FMOs are objects that move over a large distance compared to their size during the exposure time of a single frame, and possibly also rotate along an arbitrary axis with an unknown angular speed. 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 stabilization.

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

$$I_t(x) = (1 - [\mathcal{H}_t M](x))B(x) + [\mathcal{H}_t F](x)\,, \qquad (1)$$

where M is the indicator function of F. 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. 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 (1) to merge the blurred object and the partially visible background.

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,

$$[\mathcal{H}_t F](x) = \frac{1}{|P_t|}[P_t * F](x), \qquad (2)$$

	Detector	Redetector	Tracker
IN	I_{t-1}, I_t, I_{t+1}	I_{t-1}, I_t, I_{t+1} μ, r, P_{t-1}, ε	I_t, ε μ, r, P_{t-1}
OUT	μ, r, P_t	P_t	P_t
ASM	high contrast, fast movement, no contact with moving objects, no occlusion,	high contrast, fast movement, model	linear traj., model

Table 1: Inputs, outputs and assumptions of each algorithm. Image frame at time t is denoted by I_t. Symbols μ and r denote FMO mean color and radius. FMO trajectory in I_t is denoted by P_t, camera exposure fraction by ε.

where $|P_t|$ is the path length – F can then be viewed as a 2D image.

Finding all the intrinsic and extrinsic properties of arbitrary FMOs means estimating both F and $\mathcal{H}_t$, which is, at this moment, an intractable 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.[1]

We propose methods for two tasks: (i) efficient and reliable FMO localization, i.e. detection and tracking, and (ii) reconstruction of the FMO appearance, and the axis and angle of the object rotation, which requires the precise output of (i). For tracking, we use a simplified version of (1) and approximate the FMO by a homogeneous circle determined by two parameters: color μ and radius r. The tracker output (trajectory P_t and radius r) is then used to initialize the precise estimation of appearance using the full model (1).

4. Localization of FMOs

The proposed FMO localization pipeline consists of three algorithms that differ in their prerequisites and speed, we call them detector, re-detector, and tracker, and their basic properties are outlined in Tab. 1. First, the pipeline runs the fastest algorithm and terminates if a fast moving object is localized; otherwise, it proceeds to run the remaining two more complex and general algorithms. This strategy produces an efficient localization method that operates successfully in broad range of conditions. A flowchart of the whole process is shown in Fig. 4.

The first algorithm, detector, discovers previously unseen FMOs and establishes their properties. It requires sufficient contrast between the object and the background, an unoccluded view in three consecutive frames, and no interference with other moving objects. The second algorithm, re-detector, is applied in a region predicted by the FMO trajectory in the previous frames. It handles the problems of partial occlusions and object-background appearance similarity while being as fast as the detector. Finally, the tracker searches for the object by synthesizing its appearance with the background at the predicted locations. Both re-detector and tracker require trajectory information from the previous frame, initially supplied by the detector. All three algorithms require a static background or a registration of consecutive frames. To this end, we apply video stabilization by estimating the affine transformation between frames using RANSAC [10] by matching FREAK descriptors [1] of FAST features [20]. The FMO tracking is currently not robust to incorrect stabilization.

For the purpose of localization we approximate the object by a homogeneous sphere, which is typically acceptable due to fast movement and rotation. The detector updates corresponding properties, namely color μ and radius r, which are needed by the re-detector and tracker. For increased stability, the new value of any of these parameters is a weighted average of the detected value and the previous value using a forgetting function proposed in [26]. For each video sequence we also need to determine the so called exposure fraction ε, which is the ratio of exposure period and time difference between consecutive frames (e.g. 25fps video with 1/50s exposure has $\varepsilon = 0.5$). This can be done from any two subsequent FMO detections and we use average over multiple observations. We need three consecutive video frames to localize the FMO in the middle frame, which causes a constant delay of one frame in real-time processing, but this does not present any obstacle for practical use.

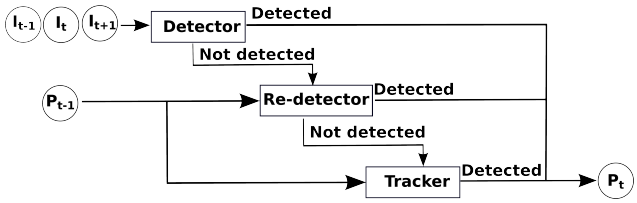


Figure 4: Flowchart of the FMO detection algorithm.

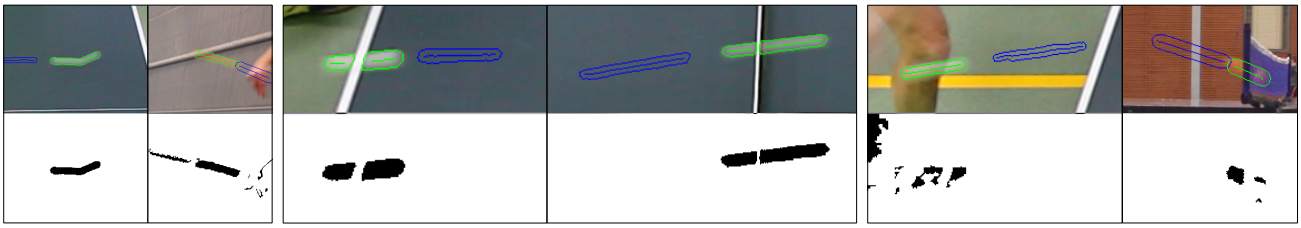
4.1. Detector

The detector is the only generic algorithm for FMO localization that requires no input, except for three consecutive image frames I_{t-1}, I_t, I_{t+1}. First we compute differential images $\Delta_+ = |I_t - I_{t-1}|$, $\Delta_0 = |I_{t+1} - I_{t-1}|$, and $\Delta_- = |I_t - I_{t+1}|$. These are binarized (denoted by superscript b) by thresholding, and the resulting images are combined by a boolean operation to a single binary image

$$\Delta = \Delta_+^b \wedge \Delta_-^b \wedge \neg\Delta_0^b. \tag{3}$$

This image contains all objects, which are present in the frame I_t, but not in the frames I_{t-1} and I_{t+1} (i.e. moving

[1] See the supplementary material for demonstration of how our method handles non-spherical objects.



(a) Detection (b) Re-detection (c) Tracking

Figure 3: (a) FMO detection, (b) redetection where detection failed because FMO is not a single connected component, (c) tracking where both algorithms failed due to imprecise Δ. Top row: cropped I_t with P_{t-1} (blue) and P_t (green) with contours. Bottom row: binary differential image Δ.

objects in I_t). A single differential image contains two responses for each FMO. When three differential images are combined in (3), multiple responses cancel out and we get only one component. In addition, the last term removes objects which did not elapse distance larger than their size during 3 consecutive frames and this way we eliminate small movement of large objects.

The next step is to identify all objects which can be explained by the FMO motion model. We calculate the trajectory P_t and radius r for each FMO candidate and determine if it satisfies the motion model. For each connected component C in Δ, we compute the distance transform to get the minimal distance $d(x)$ for each inner pixel $x \in C$ to a pixel on its component's contour. Then the maximum of such distances for each component is its radius, $r = \max d(x),\ x \in C$. Next, we determine the trajectory by morphologically thinning the pixels x that satisfy $d(x) > \psi r$, where the threshold ψ is set to 0.7. Now we decide whether the object satisfies the FMO motion model by verifying two conditions: (i) the trajectory P_t must be a single connected stroke, and (ii) the area a covered by the component C must correspond to the area $\hat{a}$ expected according to the motion model, that is $\hat{a} = 2r\,|P_t| + \pi r^2$. We say that the areas correspond, if $\left|\frac{a}{\hat{a}} - 1\right| < \gamma$, where γ is a chosen threshold 0.2. All components which satisfy these two conditions are then marked as FMOs. The whole algorithm is pictorially described in Fig. 5.

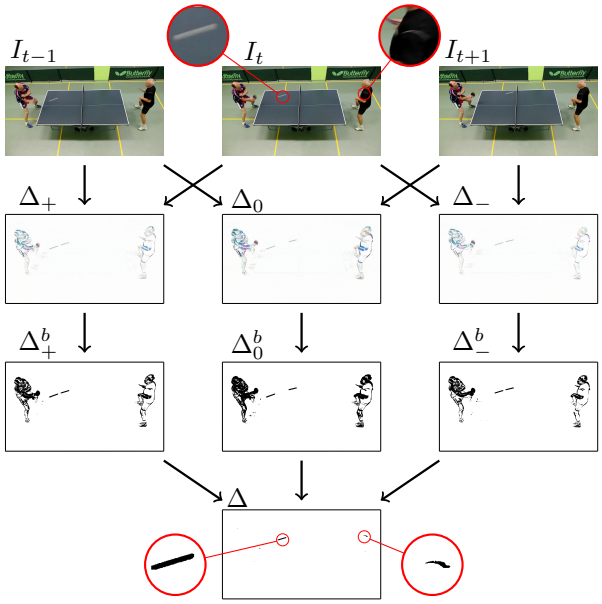


Figure 5: Detection of FMOs. Three differential images of three consecutive frames are binarized, segmented by boolean operation, and connected components are checked if they satisfy the FMO model. The two detected FMOs on this frame are the ball and the white stripe on player's t-shirt. However, only the ball passed the check and was marked as an FMO.

4.2. Re-detector

The re-detector requires previous detection of the FMO, but allows one FMO occurrence to be composed of several connected components in Δ (e.g. the FMO passes in front of background with similar color). Fig. 3 shows an example, where the re-detector finds an FMO missed by the detector. The re-detector operates on a rectangular window of a binary differential image Δ in (3), restricted to the local neighborhood of the previous FMO location. Let P_{t-1} be the trajectory from the previous frame I_{t-1}, then the re-detector works in the square neighborhood with side $4\,\frac{1}{\varepsilon}|P_{t-1}|$ and centered on the position of the previous location. Note that $\frac{1}{\varepsilon}|P_{t-1}|$, where ε is the exposure fraction, is the full trajectory length between I_{t-1} and I_t. For each connected component in this region, the trajectory P_t and radius r are computed in the same way as in the detection algorithm. The mean color μ is obtained by averaging all pixel values on the trajectory. In this region, connected components with model parameters (μ, r) are selected if the Euclidean distance in RGB $\|\mu - \mu_0\|_2$ and the normalized difference $|r - r_0|\,/r_0$ are below prescribed thresholds 0.3 and 0.4, respectively. Here, the previous FMO parameters are denoted by (μ_0, r_0).

4.3. Tracker

The final attempt to find the FMO after both the detector and re-detector have failed is the tracker, which uses image synthesis. The tracker is based on the image formation model (1) and assumes a ball-like object F with color μ and radius r moving along a linear trajectory P_t. The indicator function M is then a ball of radius r and the alpha value $A(x|P_t) := [\mathcal{H}_t M](x)$ from (1) is a simple convolution of M and P_t, (2). Using this notation, the image frame I_t can be written as

$$\hat{I}_t(x|P_t) = (1 - A(x|P_t))B(x) + \mu A(x|P_t). \quad (4)$$

The tracker now looks for the trajectory P_t that best explains the frame I_t using the approximation $\hat{I}_t$. This is equivalent to solving

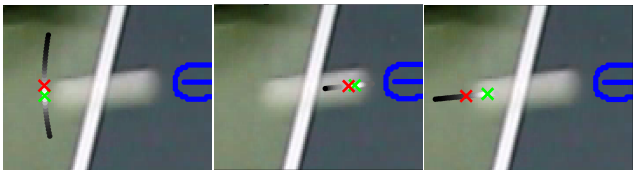
$$P_t = \arg\min_{P_t} \|\hat{I}_t(\cdot|P_t) - I_t\|_2. \quad (5)$$

As in the other two algorithms, instead of the background B we can use one of the previous frames I_{t-1} or I_{t-2}, since a proper FMO should not occupy the same region in several consecutive frames, and thus the previous frame can locally serve as the background.

A linear trajectory P_t is given by its starting point s_t, orientation β_t and length $|P_t|$ (equivalently ending point e_t). We minimize (5) over these parameters by a coordinate descent search.

First, we find the best orientation. We extrapolate the starting point linearly from the previous detection and assume that the length remains the same, $s_t = e_{t-1} + \left(\frac{1}{\varepsilon} - 1\right)|P_{t-1}|u_{\beta_t}$ and $|P_t| = |P_{t-1}|$, where $u_\beta = (\cos(\beta), \sin(\beta))$ is a unit vector with orientation β. Next we sample the space of β_t's that differ from β_{t-1} by up to 15° and choose the one that minimizes the cost (5).

The minimization w.r.t. s_t and $|P_t|$ is done in a similar manner. For s_t, we sample points in the $\frac{1}{2}|P_{t-1}|$ neighborhood of the extrapolated s_t from the previous detection, and for $|P_t|$ we again use the range $|P_{t-1}| \pm 50\%$. The three minimization stages are illustrated in Fig. 6.



(1) (2) (3)

Figure 6: Tracking steps. Detection of (1) orientation, (2) starting point, and (3) ending point. Previous detection is in blue. Green cross denotes the minimizer, red crosses the initial guess. All sampled points (gray) are scaled by their cost (5) (the darker the higher cost).

5. Estimation of appearance

Let us consider a video frame I_t acquired according to (1) and the object trajectory P_t and size r as determined by the FMO detector. The objective is to estimate the appearance F, which is a modified blind image deblurring task. One has to first estimate the blur-and-projection operator $\mathcal{H}$, and then solve the non-blind deblurring task for F. To make the estimation of $\mathcal{H}$ tractable, we limit ourselves to objects moving approximately parallel to the camera which are either spherical and rotating or arbitrarily shaped and not rotating. Function F is a representation of the object appearance – in the absence of rotation, this is directly the image of the object projected in the video frame, and when 3D rotation is present we use the spherical parametrization to capture the whole surface. In the absence of rotation, the $\mathcal{H}$ operator is convolution with the trajectory P_t as in (2) and we proceed directly to the non-blind estimation of arbitrarily shaped F.

Following the model (1) we solve the problem

$$\min_F \|(1 - [\mathcal{H}M])B + [\mathcal{H}F] - I\|_1 + \alpha\|\mathcal{D}F\|_1, \quad (6)$$

where $\mathcal{D}$ is the derivative operator (gradient magnitude) and α is the weighting parameter proportional to the level of noise in I. The L_1-norm while increasing robustness leads to nonlinear equations. We therefore apply the method of iteratively re-weighted least squares to convert the optimization problem to a linear system and use conjugate gradients to solve it. For object sizes in the FMO dataset ($r < 100$ pixels) this can be done in less than a second.

In the case of object rotation, the blur operator $\mathcal{H}$ encodes the object pose (orientation in space) as well as location in each fractional moment during the camera exposure. Trajectory aside, this is fully determined by the object's angular velocity, which we assume constant throughout the exposure. Angular velocity (in 3D) is given by three parameters (two for axis orientation, one for velocity). The functional in (6) is non-convex w.r.t the angular velocity parameters. However, we can solve it with an exhaustive search since the parametric space is not that large. We thus construct $\mathcal{H}$ for each point in the discretized space of possible angular velocities, estimate F, and then measure the error given by the functional in (6). The parametrization which gives the lowest error is the solution.

In Fig. 9 we illustrate the result of FMO deblurring in the form of temporal super-resolution. The left side (a) shows a frame captured by a conventional video camera (25fps), which contains a volleyball that is severely motion blurred. On the right side (b), the top row shows several 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. In the bottom row of

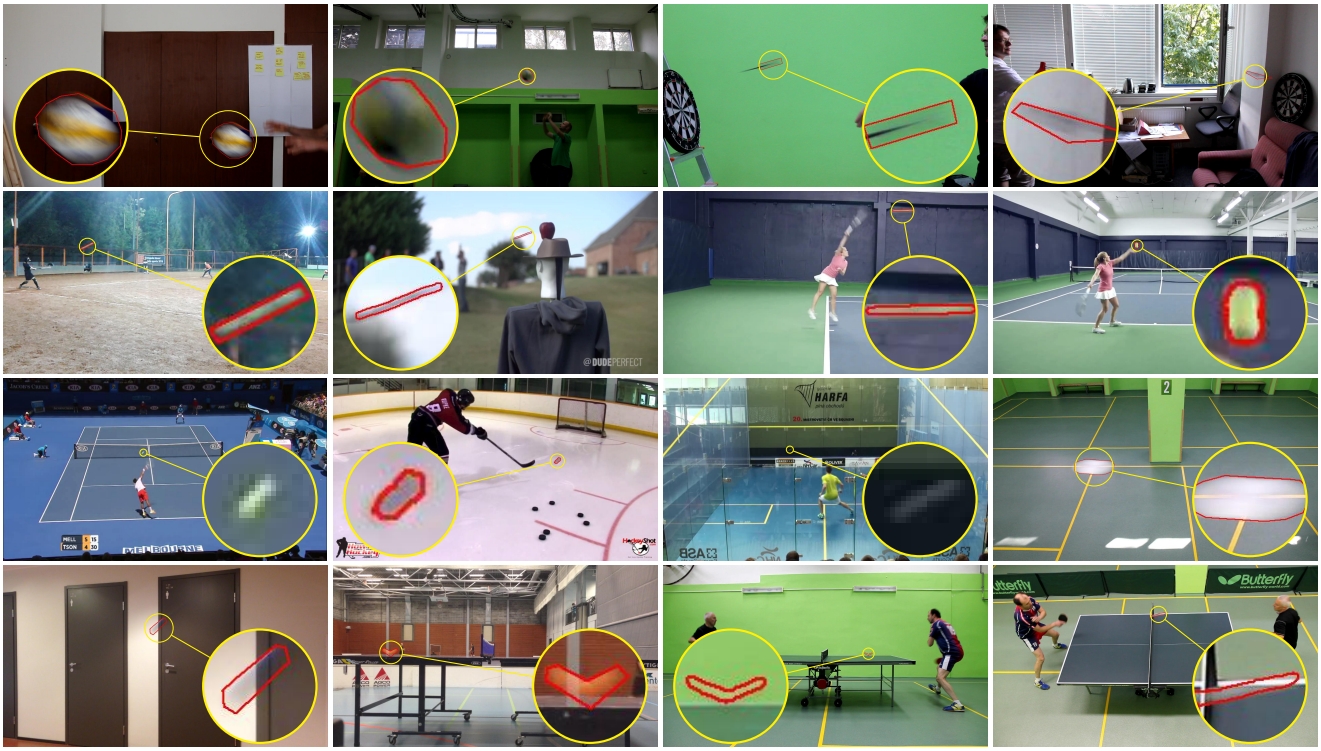


Figure 7: The FMO dataset – one example image per sequence. Red polygons delineate ground truth regions with fast moving objects. For clearer visualization two frames do not show annotations because their area consists only of several pixels. The sequences are sorted in natural reading order from left to right and top to bottom as in Tab. 2.

(b) we show the result of FMO deblurring, computed solely from the single frame in (a), 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-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 (b). 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 Fig. 10, where we use the simplified model of pure translation motion for the frisbee (bottom row).

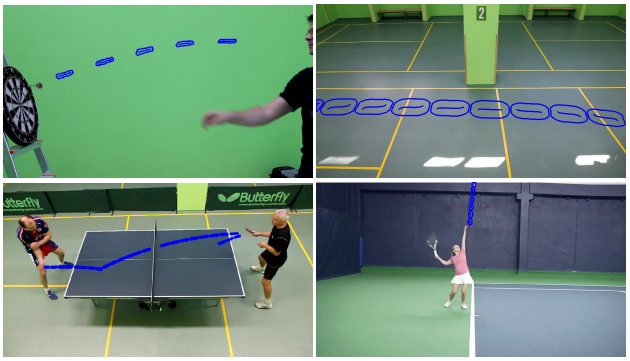


Figure 8: FMO detection and tracking. Each blue region represents object trajectory and contour in previous frames.

6. Dataset

The FMO dataset contains videos of various activities involving fast moving objects, such as ping pong, tennis, frisbee, volleyball, badminton, squash, darts, arrows, softball, as well as others. Acquisition of the videos 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 colored texture. All the 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).

None of the public tracking datasets contain objects moving fast enough to be considered FMOs – with significant blur and large frame-to-frame displacement. We analyzed three of the most widely used tracking datasets, ALOV [22], VOT [15], and OTB [27] and compared them with the proposed method 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

n	Sequence name	#	Pr.	Rc.	F-sc.
1	volleyball	50	100.0	45.5	62.5
2	volleyball passing	66	21.8	10.4	14.1
3	darts	75	100.0	26.5	41.7
4	darts window	50	25.0	50.0	33.3
5	softball	96	66.7	15.4	25.0
6	archery	119	0.0	0.0	0.0
7	tennis serve side	68	100.0	58.8	74.1
8	tennis serve back	156	28.6	5.9	9.8
9	tennis court	128	0.0	0.0	0.0
10	hockey	350	100.0	16.1	27.7
11	squash	250	0.0	0.0	0.0
12	frisbee	100	100.0	100.0	100.0
13	blue ball	53	100.0	52.4	68.8
14	ping pong tampere	120	100.0	88.7	94.0
15	ping pong side	445	12.1	7.3	9.1
16	ping pong top	350	92.6	87.8	90.1
	Average	–	59.2	35.5	40.6

Table 2: Performance of the proposed method on the FMO dataset. We report precision, recall and F-score. The number of frames is indicated by #.

Sq. name \ Method	ASMS[25]	DSST[9]	MEEM[29]	SRDCF[8]	STRUCK[12]	Proposed
volleyball	**80**	0	50	0	10	46
volleyball passing	12	6	**95**	88	8	10
darts	3	0	6	0	0	**27**
darts window	0	0	0	0	0	**50**
softball	0	0	0	0	0	**15**
archery	**5**	**5**	**5**	**5**	0	0
tennis serve side	7	0	0	0	6	**59**
tennis serve back	5	0	0	0	3	**6**
tennis court	0	0	**3**	**3**	0	0
hockey	0	0	0	0	0	**16**
squash	0	0	0	0	0	0
frisbee	65	0	6	6	0	**100**
blue ball	30	0	0	0	25	**52**
ping pong tampere	0	0	0	0	0	**89**
ping pong side	1	0	0	0	0	**7**
ping pong top	0	0	0	0	1	**88**
Average	17	1	1	1	3	**36**

Table 3: Performance of baseline methods on the FMO dataset. Percentage of frames with FMOs present where tracking was successful (IoU > 0.5).

in 91% of cases, while in the FMO dataset the displacement is 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. Fig. 2 summarizes these findings.

An overview of the FMO dataset is in Fig. 7, showing some of the included activities and the ground-truth annotations. The dataset and annotations are publicly available.

7. Evaluation

The proposed localization pipeline was evaluated on the FMO dataset. The performance criteria are precision $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, recall $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ and F-score $2\mathrm{TP}/(2\mathrm{TP}+\mathrm{FN}+\mathrm{FP})$, where TP, FP, FN is the number of true positives, false positive and false negatives, respectively. A true positive detection has an intersection over union (IoU) with the ground truth polygon greater than 0.5 and an IoU larger than other detections. The second condition ensures that multiple detections of the same object generates only one TP. False negatives are FMOs in the ground truth with no associated FP detection.

Quantitative results for individual video sequences are listed in Tab. 2. All results were achieved for the same set of parameters in the localization pipeline as discussed in Sec. 4. Performance varies widely, ranging from a F-score of 0% (complete failure) for the archery, tennis court, and squash sequences, to 100% (complete success) for the frisbee sequences. The sequences with the best results contain objects with prominent FMO characteristics, i.e. a large motion against a contrasting background. False negatives occur in three types of situations: (i) the object motion is too small (archery, volleyball), (ii) the object itself is too small (tennis court, squash), and (iii) the background is too similar to the object color (e.g., table tennis net, white edge of the table). Problem (i) can be addressed by combining the FMO detector with a state-of-the-art "slow" short-term tracker. False positives usually occur when local movements of larger objects, such as players' body parts, can be partially explained by the FMO model, or due to imprecise camera stabilization. Note that none of the test sequences contain multiple FMOs in a single frame, but the algorithm is not constrained to detect a fixed number of objects. The detection results are included in the supplementary material. Some examples are shown in Fig. 8.

FMO localization is fast, on average it takes about 20 ms. The appearance estimation step performs deconvolution, which is orders of magnitude more time consuming and depends on the FMO size. To estimate the appearance of an FMO with a 200 px diameter takes on about 1 minute (Matlab implementation).

Next, we compare the results of the FMO localization pipeline to those of several standard state-of-the-art trackers, namely ASMS [25], DSST [9], SRDCF [8], MEEM [29], and STRUCK [12]. For a fair comparison, only frames containing exactly one FMO were included. Since these

a) b)

Figure 9: Reconstruction of an FMO blurred by motion and rotation. a) Input video frame. b) Top row: actual frames from a high-speed camera (250fps). Bottom row: frames at corresponding times reconstructed from a single frame of a regular camera (25fps), i.e. 10x temporal super-resolution. The top left image shows the rotation axis position estimated from the blurred frame (blue cross) and from the highspeed video (red circle).

trackers always output exactly one detection per frame and the proposed method can return any number of detections, including none, the proposed method would have an advantage on the full set of frames. The results are presented in Tab. 3 in terms of the percentage of frames with a successful detection. Some of the standard trackers performed reasonably well on the volleyball sequences, where the motions are relatively slow, but overall results are very 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 specialized FMO tracker.

Besides FMO localization, the proposed model and estimator enable several applications which may be useful in processing videos containing FMOs. In Sec. 5 on appearance estimation, we suggested 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. The naive approach is the interpolation of adjacent frames, which is inadequate for videos containing FMOs. A more precise approach requires moving objects to be localized, deblurred, and their motions modeled, which the proposed method accomplishes (see Sec. 5), so that new frames can be synthesized at the desired frame-rate. Figs. 9 and 10 show example results of the temporal super-resolution.

Another popular use case is highlighting FMO in sport videos. Due to the extreme blur, FMOs are often hard to localize, even for humans, despite having the context provided by perfect semantic scene understanding. Simple highlighting, like recoloring or scaling, enhances the viewer's experience. Fig. 10 top-right demonstrates temporal super-resolution with highlighting.

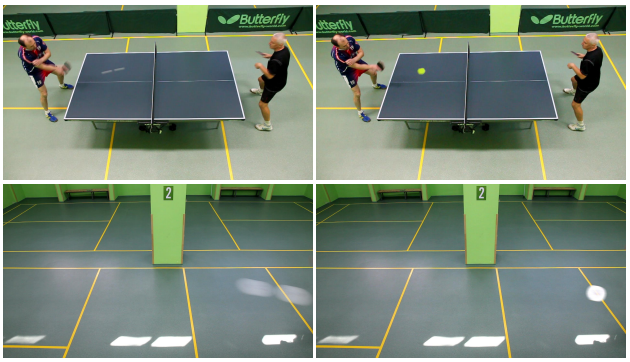


Figure 10: Temporal super-resolution using plain interpolation (left) and the appearance estimation model (right). The top right image shows the possibility of FMO highlighting.

8. Conclusions

Fast moving objects are a common phenomenon in real-life videos, especially sports. We proposed a generic, i.e. not requiring prior knowledge of appearance, algorithm for their fast localization and tracking and a blind deblurring algorithm for estimation of their appearance. We created a new dataset consisting of 16 sports videos with ground-truth annotations. Tracking FMOs is considerably different from standard object tracking targeted by state-of-the-art algorithms and thus requires a specialized approach. The proposed method is the first attempt in this direction and outperforms baseline methods by a wide margin. The estimated FMO appearance could support applications useful in sports analytics, such as realistic increase of video frame-rate (temporal super-resolution), artificial object highlighting, visualization of rotational axis and measurement of speed and angular velocity.

Acknowledgments. This research was supported by the GACR P103/12/G084 project, the Technology Agency of the Czech Republic research program TE01020415, the Czech Technical University student grant SGS17/185/OHK3/3T/13 and the GACR GA16-13830S project.

Access to computing and storage facilities MetaCentrum provided under the programme CESNET LM2015042 is greatly appreciated.

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517, June 2012. 3

[2] S. Avidan. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):261–271, Feb. 2007. 1

[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*, 33(8):1619–1632, Aug 2011. 1

[4] T. A. Biresaw, A. Cavallaro, and C. S. Regazzoni. Correlation-based self-correcting tracking. *Neurocomput.*, 152(C):345–358, Mar. 2015. 2

[5] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2512–2519, San Francisco, CA, USA, June 2010. 2

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, May 2003. 2

[7] S. Dai and Y. Wu. Motion from blur. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, June 2008. 2

[8] 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*, pages 4310–4318, 2015. 2, 7

[9] 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. 2, 7

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 3

[11] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Comput. Vis. Image Underst.*, 117(10):1245–1256, Oct. 2013. 1

[12] 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. 2, 7

[13] J. Jia. Single image motion deblurring using transparency. In *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pages 1–8, 2007. 2

[14] T. H. Kim and K. M. Lee. Segmentation-free dynamic scene deblurring. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2766–2773, 2014. 2

[15] 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. 1, 2, 6

[16] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin. A novel performance evaluation methodology for single-target trackers, Jan 2016. 1

[17] A. V. Kruglov and V. N. Kruglov. Tracking of fast moving objects in real time. *Pattern Recognition and Image Analysis*, 26(3):582–586, 2016. 2

[18] 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, Feb. 2008. 2

[19] J. Oliveira, M. Figueiredo, and J. Bioucas-Dias. Parametric blur estimation for blind restoration of natural images: Linear motion and out-of-focus. *IEEE Transactions on Image Processing*, 23(1):466–477, 2014. 2

[20] E. Rosten and T. Drummond. *Machine Learning for High-Speed Corner Detection*, pages 430–443. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. 3

[21] Q. Shan, W. Xiong, and J. 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, Oct. 2007. 2

[22] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, July 2014. 1, 2, 6

[23] J. Sun, W. Cao, Z. 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. 2

[24] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991. 2

[25] T. Vojir, J. Noskova, and J. Matas. *Robust Scale-Adaptive Mean-Shift for Tracking*, pages 652–663. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. 2, 7

[26] K. G. White. Forgetting functions. *Animal Learning & Behavior*, 29(3):193–207, 2001. 3

[27] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 6

[28] 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*, volume 3, pages 1539–1542 Vol.3, June 2004. 2

[29] J. Zhang, S. Ma, and S. Sclaroff. *MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization*, pages 188–203. Springer International Publishing, Cham, 2014. 2, 7