
Article:
Haalck, L., Mangan, M., Webb, B. et al. (2020) Towards image-based animal tracking in natural environments using a freely moving camera. Journal of Neuroscience Methods, 330. 108455. ISSN: 0165-0270

https://doi.org/10.1016/j.jneumeth.2019.108455

Towards Image-based Animal Tracking in Natural Environments Using a Freely Moving Camera

Lars Haalck[a], Michael Mangan[b], Barbara Webb[c], Benjamin Risse[a,*]

[a]*Faculty of Mathematics and Computer Science, University of Münster, Münster, Germany*
[b]*Department of Computer Science, University of Sheffield, Sheffield, United Kingdom*
[c]*School of Informatics, University of Edinburgh, Edinburgh, United Kingdom*

Abstract

Background:

Image-based tracking of individual animals can provide rich data to underpin breakthroughs in biological and medical research, but few if any existing methods extend to tracking unconstrained natural behaviour in the field.

New method:

We have developed a visual tracking system for animals filmed with a freely moving hand-held or drone-operated camera in their natural environment. This exploits a global inference method for detecting motion of an animal against a cluttered background. Trajectories are then generated by a novel video keyframe selection scheme in combination with a geometrically constrained image stitching algorithm, resulting in a two-dimensional panorama image of the environment on which the dense animal path is displayed.

Results:

By introducing a minimal and plausible set of constraints regarding the camera orientation and movement, we demonstrate that both per-frame animal positions and overall trajectories can be extracted with reasonable accuracy, for a range of different animals, environments and imaging modalities.

Comparison:

*Corresponding author
Email addresses: lars.haalck@uni-muenster.de (Lars Haalck), m.mangan@sheffield.ac.uk (Michael Mangan), bwebb@inf.ed.ac.uk (Barbara Webb), b.risse@uni-muenster.de (Benjamin Risse)

Our method requires only a single uncalibrated camera, does not require marking or training data to detect the animal, and makes no prior assumptions about appearance of the target or background. In particular it can detect targets occupying fewer than 20 pixels in the image, and deal with poor contrast, highly dynamic lighting and frequent occlusion.

Conclusion:

Our algorithm produces highly informative qualitative trajectories embedded in a panorama of the environment. The results are still subject to rotational drift and additional scaling routines would be needed to obtain absolute real-world coordinates. It nevertheless provides a flexible and easy-to-use system to obtain rich data on natural animal behaviour in the field.

Keywords: animal tracking; visual tracking; outdoor field experiments; panorama stitching; projective geometry

---

1. Introduction

Recent advances in automatic image-based tracking of individually behaving animals have enabled the collection of rich datasets underpinning breakthroughs in biological and medical research [1]. Parallel improvements in imaging technology, computational power and computer vision algorithms have supported implementation of novel tracking systems for behavioural analysis of organisms ranging from tiny insects (e.g. Drosophila melanogaster) to larger vertebrates (e.g. mice; for a review see [2]). Yet, such systems have been largely developed for controlled laboratory conditions and struggle to generalise to real-world situations [3] preventing tracking of animals in their natural environments [1]. The importance of in-field behavioural analysis arises in many contexts, including the influence of fertilisers on navigation capabilities [4], the impact of factory farming on animals regulating greenhouse gas emissions [5], and the threats of light pollution on biodiversity in general [6]. Most behavioural quantifications for these studies are still done manually; the need for novel automatic methodologies is emphasised in multiple publications [1, 7, 8, 9].

Challenges of in-field tracking include (but are not limited to): potentially very small or varying animal sizes; changing animal appearances; clutter and occlusions; limited number of recordings; varying illumination and shadows; and an unknown, potentially unlimited, environment and spatial range through which the animal may move [1]. Addressing these challenges requires a robust detection and tracking algorithm which relies on as few constraints as possible. For example, a freely moving camera is required to capture animal paths in arbitrarily sized environments. Existing machine learning methods for detection may not be applicable due to a lack of training data (few recordings) or low resolution of the animal in the video image preventing effective discriminative correlation [10]. Finally, long trajectories extracted from a moving camera will inevitably suffer from drift and thus error accumulation [11].

An ideal image-based tracking system for biologists should be applicable to a diversity of species, not require any animal tagging or marking, and function within a wide range of environments and experimental conditions. Furthermore, it should be mostly automated, simple to use, and inexpensive [1]. For practicality in field studies, it would be desirable if only a monocular freely moving hand-held or drone-operated camera was required to record the movement of the animal. Ideally, no additional sensor information (such as inertial measurements from an accelerometer, distance sensing from line of flight or stereo) should be needed, and camera calibration should not be a requirement if it is desired to enable processing of already existing videos and handle arbitrary zooms during recordings.

In this work we introduce a general visual tracking approach for animals moving freely in their natural environments recorded with a freely moving camera. The system has two parts: (1) a universal detection mechanism to localise the position of the animal in each video frame and (2) a tracking mechanism to extract camera motion-compensated animal trajectories over time. Building on our robust and globally optimised detection framework [12] we are able to localise even tiny and low contrast objects like insects in cluttered high-resolution images. We extend this system using a novel video key-frame selection mech-

anisms to extract a subset of frames to generate a 2D panoramic image of the underlying scene. Subsequently, geometrically constrained dense animal trajectories are projected into the panorama to visualise the entire track. For our real world experiments we chose small insects (ants, dung beetles and woodlice) recorded with different hand-held devices, and also tested the method on a video of mammals (wild dogs) recorded by a drone in Africa.

1.1. Related Work

To date behavioural quantifications of animals in natural environments have mainly been done using non-visual techniques like telemetry [7]. These methods have limited applicability due to the need to tag the animals with sensors, which is possible only for a fraction of species [8]; and tags may crucially affect the behaviour [13]. Furthermore, telemetry has a limited temporal resolution and hence does not easily reveal the animal's actions, nor does it provide any information about the surrounding environment [1] which is often crucial for interpreting behaviour.

In contrast, image-based tracking enables high temporal resolution and provides the visual context. Computer vision and machine learning have lately achieved remarkable tracking accuracies in many contexts including medicine, surveillance and autonomous navigation. Particularly, deep learning algorithms have improved the accuracy of visual object tracking, as reflected in the yearly visual object tracking (VOT) evaluation of more than 50 different tracking systems [14]. Top performing algorithms rely on correlation filtering, using convolutional neural networks (CNNs) as feature extractors, and examples of successful application to animal tracking include automatic detection of marine species in aerial imagery [15] and identification of individual animals in crowded collectives under laboratory conditions [16] (a more complete survey of animal tracking can be found in [1]).

However, correlation filtering approaches require a minimum target resolution and fail for instances of small animals occupying only a few pixels [17]. In addition, it has recently been shown that increased background heterogene-

ity and amplified coupling between a few animals and the background drastically degrade the performance of state-of-the-art machine learning based trackers [18]. As a consequence, a key animal tracking scenario remains relatively unaddressed, namely, visual tracking of animals in their natural habitat [1].

2. Methods

We address the problem of tracking animals in their natural habitat by using a novel tracking strategy that imposes only four constraints on the video capture: (1) The animal has to move in more than 50% of the frames and the motion has to be roughly equally distributed over the video sequence; (2) the frame rate has to be fast enough to ensure relatively small displacements between consecutive frames (no more than a few body lengths); (3) the background has to have enough distinctive texture to allow feature-based image warping; and (4) the imaging plane of the camera should be kept parallel to the ground (i.e. bird's eye view) and the distance to the ground should not vary strongly. Note that a violation of the last constraint will not prevent our algorithm from tracking but will result in scaling issues in the final trajectory (see below). Our method imposes no constraints on the animal's size or appearance, nor on the appearance of the environment, and does not require any labelling or learning. The resulting system is robust against varying illumination, shadows, clutter and occlusions, and can be directly applied to already existing videos, as no camera calibration is required. Given a moving camera, the algorithm will automatically select whichever animal has the dominant motion with respect to the entire video, so that no manual initialisation is required. A single animal will thus be tracked consistently and motion cues from other animals or distracting items will be reliably discarded.

As illustrated in Figure 1 (A) the algorithm assumes a birds-eye camera (image plane is parallel to the ground) hovering above the animal of interest. Assuming constant height (z), the camera can be moved in the x,y directions and rotated around its optical axis (i.e. z axis; cf. constraint 4). In our first

processing step we extract the motion of the camera between consecutive frames using ORB features [19] in combination with a robust estimator (i.e. RANSAC) to calculate the full perspective transformation H_{t+1}^{t} between frames at $t+1$ and t (called $f(t+1)$ and $f(t)$; cf. constraint 3; Figure 1 (B)). These consecutive transformations and ORB features are subsequently used in the detection and tracking routines as described below.

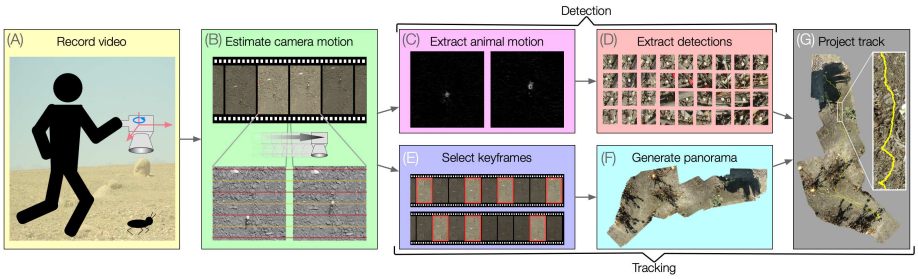

Figure 1: Overview of our imaging and tracking system. **(A)** Recordings are made using a hand-held camera assuming translations along the x and y and rotation around the z direction. **(B)** The algorithm estimates the camera motion between consecutive frames. **(C)** For the animal detection the camera motion is removed and the remaining motion is used as an indicator for potential animal positions (i.e. unaries). **(D)** Per-frame locations are extracted from the unaries using global optimisation [12]. **(E)** In parallel a sparse set of key-frames are extracted based on a heuristically constrained forward search. **(F)** These key-frames are used to generate a panorama. **(G)** Finally, in-frame detections from the entire sequence are projected into the panorama to generate a dense animal trajectory (given in yellow).

2.1. Detection

We can warp the camera position at $f(t+1)$ on the position at $f(t)$ by using a perspective transformation H_{t+1}^{t} resulting in a new virtual frame

$$\tilde{f}(t+1) = H_{t+1}^{t} \cdot f(t+1).$$

The virtual frame $\tilde{f}$ at time $t+1$ appears to be at the same location as its reference frame $f(t)$ giving the impression that the camera is stationary between these images. Therefore, frame differencing between $f(t)$ and $\tilde{f}(t+1)$ can be used to extract the remaining motion which consists of the animal motion and

noise $u(t) = |f(t) - \tilde{f}(t+1)|$ (cf. constraint 1). The difference image $u(t)$ (called a unary) is a 2D heat map where bright pixels indicate the remaining motion (Figure 1 (C)). These cues are treated as observed variables which are combined with a smooth motion model (cf. constraint 2) to formulate an inference problem. Smooth motion is implemented by using small (i.e. low variance) 2D Gaussian distributions centred at the hypothetical animal position at time t, called pairwise potentials. Both the unaries and the pairwise potentials are combined in a 'factor graph' which is a graphical representation of the underlying inference problem. Animal detections are then calculated by extracting the maximum value over all unaries where consecutive unaries are smoothly connected by the pairwise potentials. This is globally optimised by using the max-sum algorithm (for details of the detection algorithm see [12]). As a result we get detections $p_t \in \{(x,y)|x,y \in \mathbb{N}\}$ ($t = 1,...,T$; T is the total number of frames) specifying the (x,y) position of the animal in each frame t in pixel coordinates (Figure 1 (D)).

2.2. Tracking

Detections only specify 2D animal positions in the respective frame coordinate system, so in order to extract movement trajectories, the detections have to be warped relative to a single reference frame by reusing the image transformations. However, the concatenated use of consecutive transformations over long sequences will inevitably accumulate an error (i.e., drift) and result in tracks with a decreasing accuracy over time [11]. Therefore, we use a three-stage procedure to extract panorama images featuring the animal track with manageable drift: (1) key-frame selection (Figure 1 (E)), (2) panorama generation (Figure 1 (F)) and (3) animal trajectory projection (Figure 1 (G)).

Key-frame selection. We first extract panorama images covering the ground plane of the entire video sequence based on a strongly reduced subset of all available frames, called key-frames $f(\tau_i)$ with $\tau_i \in I^{key} \subset \{1,2,...,T\}$. The key-frame selection is implemented as a forward search algorithm using two opposing

heuristics: a requirement for sufficient frame-to-frame overlap; and a requirement for a small total number of key-frames. The former heuristic ensures a good coverage of the underlying scene whereas the latter aims to reduce accumulative drift and computing load. Given an already selected key-frame $f(\tau_{i-1})$ we use a window of size 50 to search for an appropriate successor $f(\tau_i)$. Within this window the overlap is quantified by the total number of shared ORB matches assuming uniform feature point distributions. As demonstrated in Figure 2 (orange plot) this quantity usually decreases with an increasing successor index τ_i but not monotonically due to the overall quality of the extractable features of frame $f(\tau_i)$. However, the direct successor $f(\tau_{i-1}+1)$ will inevitably have most matches so that the second heuristic has to ensure larger shifts between $f(\tau_{i-1})$ and $f(\tau_i)$ leading to smaller sets I^{key}. The shift is calculated as the L2 norm of the median shift over all matched feature points (Figure 2 blue plot) and constrained by a lower and upper bound t_{lower} and t_{upper}. Given a video resolution $N \times M$, let $m = \min(N, M)$. We then calculate $t_{lower} = \frac{1}{4}m$ and $t_{upper} = \frac{3}{4}m$ providing a minimal shift of at least 25% and a maximal shift of at most 75% (dashed lines in Figure 2). If both heuristics are applied the chosen key-frame $f(\tau_i)$ in boundaries $[t_{lower}, t_{upper}]$ is the frame with highest amount of geometrically verified matches and becomes the next reference frame (Figure 2 green line). If no frames in the current window satisfy the shift constraint the last frame of the current window is selected as the new key-frame.

Panorama generation. The resultant key-frames are used to generate a panorama image covering the entire video. By using the matches determined during key-frame selection we calculate the transformation $T_{\tau_i}^{\tau_{i+1}}$ mapping key-frame $f(\tau_i)$ to key-frame $f(\tau_{i+1})$. Unfortunately, standard panorama extraction algorithms cannot directly be used for mainly two reasons: Firstly, most panorama generation algorithms assume a rotation around their optical centre of the camera [20]. In our imaging scenario, the camera scans across the environment (characterised by translational motion parallel to the ground). Secondly, panoramas from videos inevitably suffer from accumulative drift since thousands of frames need

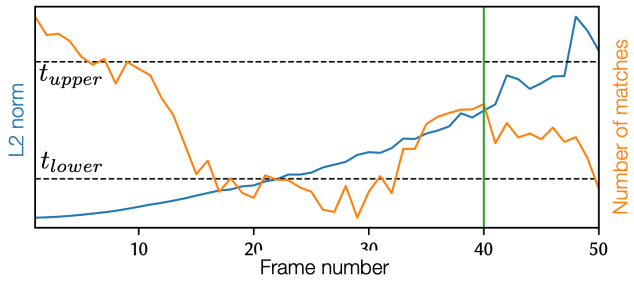

Figure 2: Key-frame selection algorithm. Using a forward search of 50 consecutive frames we quantify the Euclidean distance (L2 norm in blue) of the shifted median feature point movement based on the perspective transformations from the camera motion estimation (distance in pixel and constrained by a lower and upper boundary given as dashed lines called t_{lower} and t_{upper}). In addition, we keep track of the number of shared features from the reference frame to the 50 successors (number of shared matches given in orange). The next key-frame is determined by estimating the best trade-off between distance travelled and number of matches (green line).

to be warped [11]. The potential directions of drift correlate with the degree of freedom (DoF) of the used transformation $T_{\tau_i}^{\tau_{i+1}}$, whereas the DoF directly correlates with camera motion (cf. Figure 3). Assuming camera motion in all directions (translation and rotation) affine and projective transformations are required (the latter also incorporates perspective geometric changes which result from the 2D projection of the 3D environment). Even if only the sparse set of key-frames is used, affine transformations suffer severely from scale drift and projective transformations often collapse due to perspective drift (Figure 3 (C) and (D)). Furthermore, both transformations induce shearing of the flat surface geometry. However, assuming the camera plane is parallel to the ground, such shearing can only be induced by rotations around the y and x axis. By eliminating these rotations, the transformation $T_{\tau_i}^{\tau_{i+1}}$ is reduced to a similarity transformation. As visible in Figure 3 (B) neither shearing nor perspective artefacts perturb the resultant panorama. However, translations along the z axis (i.e. changing the distance between the image and ground plane) induce scale drift.

We therefore restrict our transformation model as described by constraint (4) above, that is, we assume there are only translations along the x and y axis, and only rotations around the z axis. Thus $T_{\tau_i}^{\tau_{i+1}}$ can be implemented as an isometry (Figure 3 (A)). The panorama $\mathcal{P}$ is thus calculated by warping consecutive pairs of key-frames from $\tau_i, \tau_{i+1} \in I^{key}$ using the isometry $T_{\tau_i}^{\tau_{i+1}}$ followed by plain image stitching (we do not use advanced blending in order to identify image borders for quality checks).

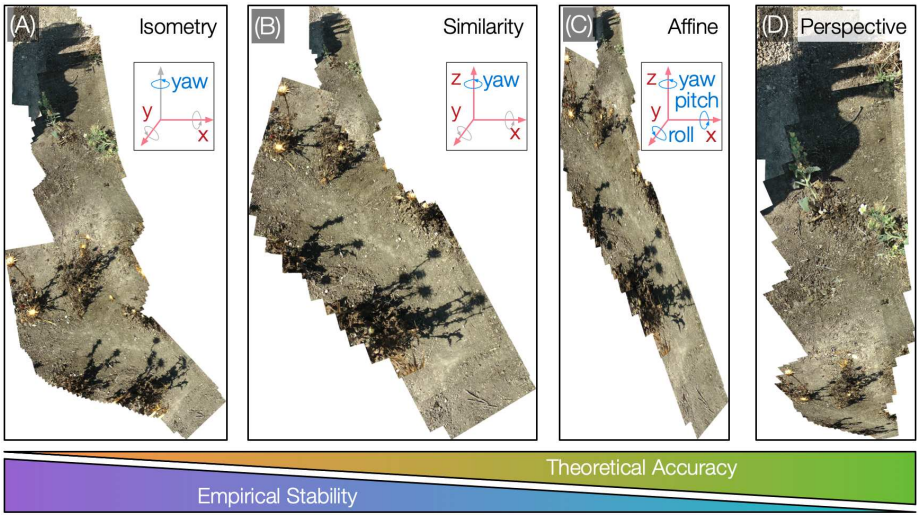

Figure 3: Trade-off between empirical stability and theoretical accuracy as additional degrees of freedom are included in panorama generation (clutter scenario from Table 1; compare to Figure 5 (A)). **(A)** Isometry assumes only translation along the x and y axis and rotation around the z-axis (for a downward-facing camera). **(B)** Similarity also compensates for translation along the z-axis (note however the scale drift at the top side of the panorama). **(C)** Affine and **(D)** perspective transformations allow all axis translations and rotations, but the reconstructions tend to collapse for longer videos due to drift-induced instabilities. Thus for practical purposes, Isometry (A) and Similarity (B) yield better results, although theoretically less accurate.

Trajectory projection. The final step is to combine the detections p_t specifying the (x, y) position of the animal in frame coordinates at time t, and the panorama image $\mathcal{P}$ generated using isometries $T_{\tau_i}^{\tau_{i+1}}$ for consecutive key-frames at time τ_i and τ_{i+1} and transformations H_t^{t+1} for all consecutive frames

$t \in \{1, ..., T\}$. In order to project the animal trajectory into the panorama, all in-frame positions p_t need to be transformed relative to the first reference frame $f(1)$. Given the key-frame transformations $T_{\tau_i}^{\tau_{i+1}}$, a sparse trajectory can be calculated by

$$\tilde{p}_{\tau_j} = \left(\prod_{i=1}^{j-1} T_{\tau_i}^{\tau_{i+1}} \right) \cdot p_{\tau_j} \tag{1}$$

where $\tilde{p}_{\tau_j}$ is the warped animal position in the panorama $\mathcal{P}$ at key-frame τ_j and $\left(\prod T_{\tau_i}^{\tau_{i+1}}\right)$ accumulates all isometries multiplicatively. The resultant sparse trajectory is given in Figure 4 (cyan track). However, in order to have a more regular and dense trajectory, all animal detections p_i need to be projected onto the panorama. This is done by using dense isometries T_t^{t+1} for all consecutive frames and again warping animal detections by multiplicative forward projections

$$\tilde{p}_t = \left(\prod_{i=1}^{t-1} T_i^{i+1} \right) \cdot p_t \tag{2}$$

Note that due to the increase in multiplications the cumulative error caused by drift is also increased. The resultant animal positions $\{p_1, \tilde{p}_2, \tilde{p}_3, ..., \tilde{p}_T\}$ are the final dense and regular trajectory until the last frame T in coordinates of the reference frame $f(1)$ as shown by the yellow track in Figure 4.

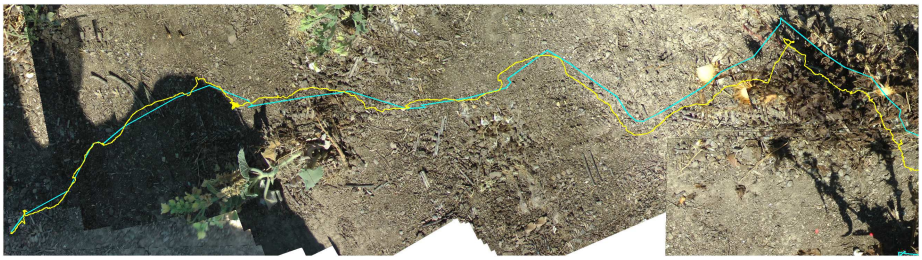

Figure 4: Trajectory projection example (clutter scenario from Table 1; see also Figure 5 (A)). The cyan line represents the sparse trajectory based on the key-frames only and the yellow line represents the dense track from all frames.

3. Results

We evaluated our tracking algorithm using 6 different videos featuring strongly varying imaging and environmental conditions. The overall characteristics of these videos are given in Table 1. The column titled Time states the approximate total running time (in minutes) to process the respective video on a machine with a 14 core processor (Intel i9-7940X) and no GPU.

Video	**Frames**	**Key-frames**	**Resolution**	**Animal size**	**Time**
clutter	2500	138 (5.52%)	1920×1040	44	≈ 34
occluded	750	45 (6%)	1920×1040	36	≈ 10
beetle	3357	117 (3.49%)	2704×1440	56	≈ 63
mobile	600	17 (2.83%)	1920×1080	13	≈ 8
nightvision	840	28 (3.33%)	1920×1080	98	≈ 13
drone	1677	63 (3.76%)	960×720	82	≈ 21

Table 1: Evaluation dataset overview. Six different videos featuring different animals and environments and captured with different cameras are used for evaluation. Resolution and animal size (diameter) is given in pixels. Key-frames are given in absolute numbers and relative to the number of frames. Computation time is given in minutes. 'clutter' and 'occluded' were recorded using a camcorder, 'beetle' was recorded using a GoPro, 'mobile' was recorded using a mobile phone, 'nightvision' was recorded using a night vision camera and 'drone' was captured by a drone operated camera.

The 'clutter' video shows an ant navigating in highly cluttered terrain including occlusions and moving shadows. The 'occluded' video features a very small ant (36 pixel diameter in a 1920×1040 video) which is occluded in more than 100 frames in a very irregular and dynamic environment. In contrast, the 'beetle' video features an easy to recognise dung beetle target, without clutter or occlusion, but is compromised by the moving shadow of the camera operator, wide angle recording, abrupt camera motions and a dung ball being moved by the beetle. The 'mobile' video shows a woodlouse recorded with a mobile phone in an urban environment. This video has very low animal resolution (13 pixels in a full HD video) in combination with a very low animal-background contrast

and strong jitter. The 'nightvision' video is recorded using a high resolution night vision camera to image nocturnal ants resulting in noisy images (due to high ISO settings) and false colours. Finally, the 'drone' video is recorded using a drone and features multiple targets (wild dogs) in a visually sparse environment (the animal which appears in the centre of the image in most frames was automatically selected as the tracking target).

In 'clutter', 'occluded', 'mobile', 'nightvision' and 'drone' video we kept the image plane approximately parallel to the ground (bird's eye view), but the 'drone' video violates the constant height assumption of constraint 4 by changing the drone's altitude in the middle of the recordings. Furthermore, the 'beetle' video includes severe rapid rotations and translations in all directions, thus also violating constraint 4. We note that constraint 2 was not crucial in any of these videos since the standard frame rates of 30 to 60 fps are usually sufficient to ensure small displacements of the animal between consecutive frames. As a consequence we could use a fixed sigma value for the 2D Gaussian potentials that weight the expected displacement in consecutive frames, equivalent to several body lengths of the animal.

Furthermore, the displacement between consecutive frames is only weighted by 2D Gaussian potentials and we used a fixed sigma value covering several body lengths of all animals.

Since the overall goal of our study was to identify the constraints, possibilities and limitations of in-field animal tracking using a freely moving camera our evaluation comprises qualitative and quantitative results. A detailed evaluation of the detection accuracy is given elsewhere [12]. Therefore, we focus on the panorama and trajectory generation here.

3.1. Qualitative evaluation

An overview of the qualitative results is given in Figure 5 (white arrows and 'start' and 'end' markers indicate the movement direction of the animal). The panorama stitching algorithm managed to generate realistic background pictures for all scenarios The most artefacts are seen in the beetle video, which

is mainly caused by violating constraint 4: erratic motion can be recognised in all directions. In addition, the extreme wide angle of the GoPro camera induces stitching artefacts. It is notable that in videos covering long distances (clutter, nightvision and drone) our algorithm extracted accurate panoramas based on a strongly reduced set of key-frames (Table 1). Sparse trajectories (blue lines in Figure 5) indicate the exact animal positions for the keyframes in the resultant panorama. In contrast, dense trajectories (yellow lines in Figure 5) show more motion details and are sampled more regularly. In four of six scenarios the drift of the dense track, relative to the sparse track, remains within reasonable bounds for the majority of the track. Only in the beetle and the drone scenario a stronger divergence can be observed over time (see quantitative evaluation below). Note that the drift of the dense trajectory increases towards the end of the movement path in all scenarios.

3.2. Quantitative evaluation

An in-depth evaluation of the detection accuracy can be found in [12], in which we used a publicly available (Small Target within Natural Scenes; STNS) dataset [21] to benchmark our algorithm. In summary, we measured the detection accuracy as the distance of the detection to the manually specified target position in normalised animal lengths. The average distance of the detection to the centre of mass was below 0.36 animal lengths in the STNS dataset and the first and third quantile is 0.27 and 0.52 respectively [12].

Given these highly accurate detections in each frame and qualitatively reasonable panoramas we will here focus on the accuracy of the dense track in comparison to the sparse tracks. Dense trajectories resulting from consecutive frames (cf. Equation 2) are more useful for behavioural quantifications since they represent regular position estimates in contrast to the key-frame based sparse estimations (cf. Equation 1). However, due to the higher number of multiplicative isometry transformations there is also more translational and rotational drift. Therefore, we quantify the drift as the Euclidean distance between the sparse animal positions and the dense positions over time (Figure 6). In the

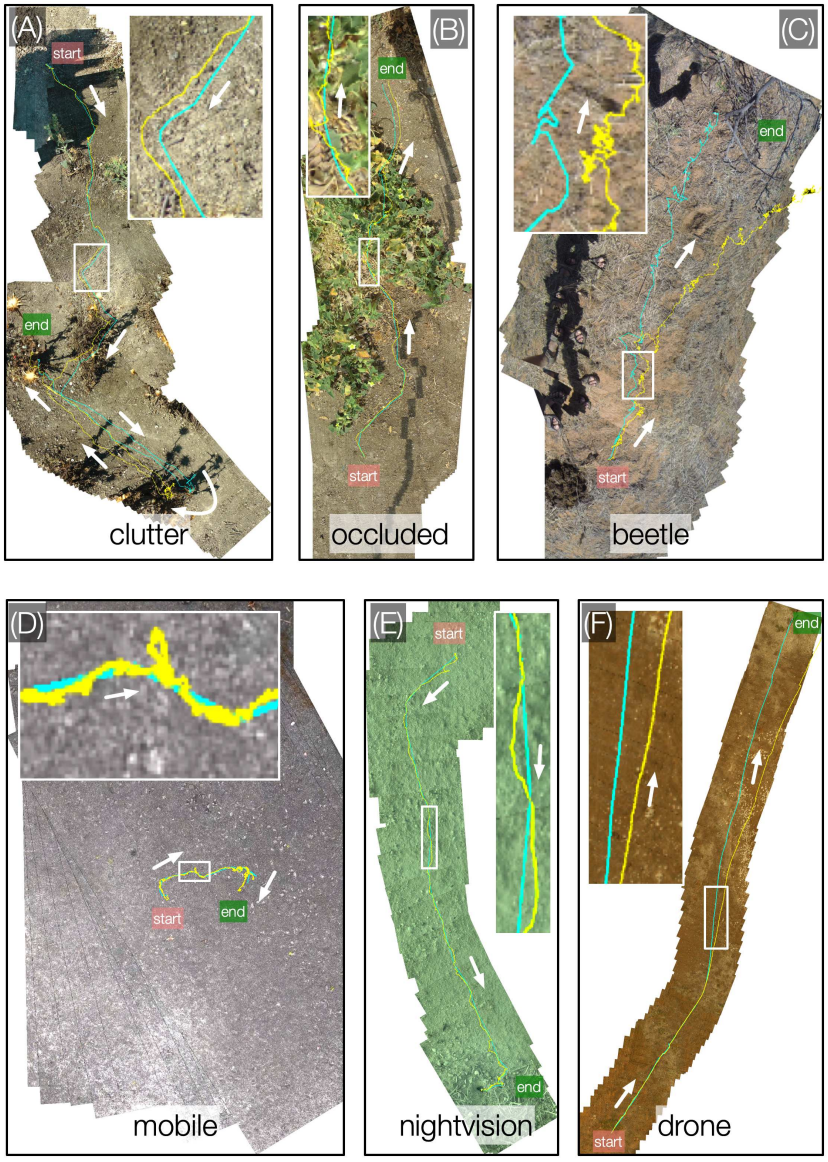

Figure 5: Qualitative evaluation. Panoramas including the sparse (cyan) and dense (yellow) trajectories for all test videos (cf. Table 1): **(A)** clutter; **(B)** occluded; **(C)** beetle; **(D)** mobile; **(E)** nightvision; and **(F)** drone. Insets show a close-up of the trajectory segment indicated by the white rectangle. Arrows indicate the movement direction.

occluded, mobile and night vision scenario the drift remains within reasonable boundaries. Since the length of the sparse and dense tracks do not differ strongly the rotational drift induces most of the error as also visible in the beetle scenario in Figure 5 (C). The occluded video manages to maintain low drift until key-frame 22 and starts to diverge after this frame. Due to error propagation of a single erroneous rotation the error will inevitably increase over time. For the recordings that adhere most closely to our four constraints (occluded, mobile and nightvision) the resultant panoramas and dense trajectories appear to provide a good estimate of the actual behaviour.

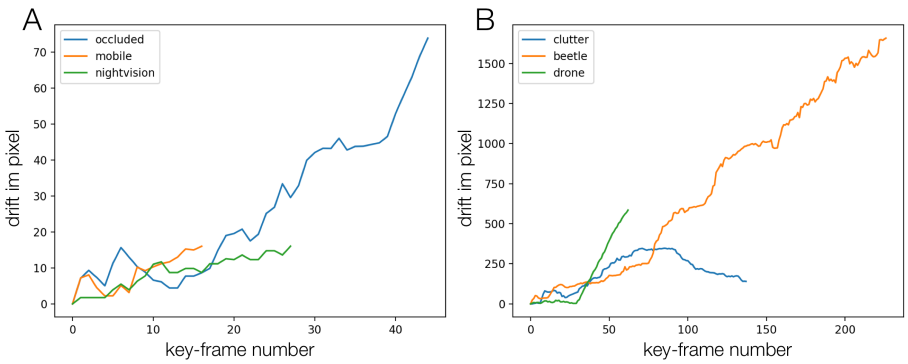

Figure 6: Drift evaluation. Drift between sparse and dense trajectories is given for each key-frame (cf. Table 1). (A) Results for the occluded, mobile and nightvision scenario. (B) Results for the clutter, beetle and drone scenario. Note the difference in scaling.

To show how the extracted tracks allow estimation of quantitative behavioural features, we plot the velocity of the ant from the clutter video in Figure 7. The raw velocity is depicted in light blue and a smoothed velocity is drawn in dark blue. The latter was calculated by extracting the mean over time using a sliding window of size 50. Note that due to the use of isometries, unintended scale shifts, resulting from translations in z direction, are not compensated and will result in domain shifts of the velocity (constraint 4). However, stop phases and trends in velocity can easily be recognised.

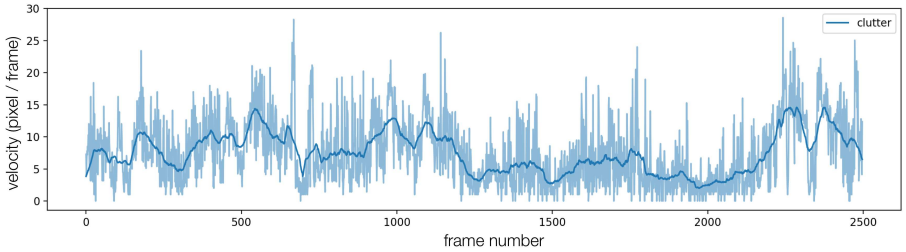

Figure 7: Velocity evaluation. Velocity for video clutter is given in $\frac{pixel}{frame}$ (cf. Table 1). The light blue line indicates the raw measurements and dark blue represents the smoothed velocity measure.

4. Discussion

Image-based tracking of animals in their natural environment is a challenging and as yet unsolved problem [1]. In particular, the complex appearance of the fore- and background as well as frequent lighting and other disturbances prevent the usage of techniques developed for well controlled laboratory situations. We have developed a tracking prototype to support the extraction of position information from videos of animals shot with a moving camera in the field. Our algorithm imposes only four basic constraints, is not limited by the appearance or resolution of the target and does not require any training. As a consequence, the algorithm is not limited to animals or natural environments and can be applied to all kinds of moving objects and scenes (an examples of correctly detected artificial objects in urban environments is given in [12]).

Whereas we have shown a reliable detection of in-frame positions is possible in many different situations [12] a reliable camera-motion compensated trajectory extraction remains an open challenge. Advanced trajectory generation is required to cope for the drift which will inevitably occur in case of visual camera tracking [11]. Since no existing method is available to benchmark our algorithms directly [1] we plan in future to use a motion capture system (following the camera) to evaluate our algorithms in more detail.

The focus of our study was to track a single small object in a potentially cluttered environment using a moving camera. The animal size in the videos

varied between 13 and 98 pixels, limiting the possibility to extract any information other than overall position in the environment. Extending the method to obtain pose information (for example see [22]) or to extract visually distinct features to support multi-animal tracking [16] would require filming in which the animal occupied more pixels, but should otherwise be straightforward.

This preliminary study does not claim to produce reliable animal tracks in real-world coordinates. For example, rotational drift will inevitably induce errors over time resulting in incorrect heading directions and thus erroneous trajectories after wrong rotations. In addition, the isometries used here for panorama generation discard rotations around the x and y axis as well as translations along the z-axis. Moreover, the resultant trajectory is in camera coordinates and only relative to the reference frames so that additional scaling routines are required to achieve absolute real-world measurements.

Nevertheless, we have made significant progress towards this goal. Since tracking results are difficult to interpret (especially in situations in which a moving camera is used) we implemented a rudimentary GUI to generate and inspect trajectories manually. This allows errors in panoramas and dense tracks to be easily spotted. Provided the constraints are not violated, our algorithm is capable of extracting insightful qualitative trajectories embedded in a panorama showing the overall environment. We also note that our algorithm can directly be applied to already existing videos from all kinds of imaging devices since no additional hardware nor calibration is necessary. Our work has particular relevance for insect researchers, as it is effective for tiny animals in complex environments, and overcomes the substantial limitations of telemetry [13, 8] for insect monitoring. With additional research, we believe it will soon be possible to offer biological researchers a complete, flexible, easy-to-use tool for tracking of animals in their natural environments.

Acknowledgement

The authors would like to thank all groups providing videos for our study. In particular we would like to thank Marie Dacke (University of Lund), Stephen Lang and Hemal Naik (MPI Konstanz), Huw Cordey (Silverback Films) and Ajay Narendra (Macquarie University). This project is funded by the BBSRC, the EPSRC (EP/M008479/1) and the Microsoft AI for Earth initiative. Lars Haalck would like to acknowledge his funding from the Heinrich Boell Stiftung.

References

[1] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. de Polavieja, L. P. J. J. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, U. Brose, Automated image-based tracking and its application in ecology, Trends in Ecology & Evolution (2014).

[2] S. E. R. Egnor, K. Branson, Computational Analysis of Behavior, Annual Review of Neuroscience (2016).

[3] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual Tracking - An Experimental Survey., IEEE Trans. Pattern Anal. Mach. Intell. (2014).

[4] J. Fischer, T. Müller, A.-K. Spatz, U. Greggers, B. Grünewald, R. Menzel, Neonicotinoids Interfere with Specific Components of Navigation in Honeybees, PloS ONE (2014).

[5] T. J. Hammer, N. Fierer, B. Hardwick, A. Simojoki, E. Slade, J. Taponen, H. Viljanen, T. Roslin, Treating cattle with antibiotics affects greenhouse gas emissions, and microbiota in dung and dung beetles., Proceedings of the Royal Society of London B: Biological Sciences (2016).

[6] F. Hölker, C. Wolter, E. K. Perkin, K. Tockner, Light pollution as a biodiversity threat, Trends in Ecology & Evolution (2010).

[7] R. Kays, M. C. Crofoot, W. Jetz, M. Wikelski, Terrestrial animal tracking as an eye on life and planet., Science (2015).

[8] W. D. Kissling, Animal telemetry: Follow the insects, Science (2015).

[9] R. Dirzo, H. S. Young, M. Galetti, G. Ceballos, N. J. B. Isaac, B. Collen, Defaunation in the Anthropocene., Science (2014).

[10] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg, Convolutional Features for Correlation Filter Based Visual Tracking., ICCV Workshops (2015).

[11] H. Strasdat, J. M. M. Montiel, A. J. Davison, Scale Drift-Aware Large Scale Monocular SLAM., Robotics Science and Systems (2010).

[12] B. Risse, M. Mangan, L. Del Pero, B. Webb, Visual Tracking of Small Animals in Cluttered Natural Environments Using a Freely Moving Camera (2017).

[13] T. McIntyre, Animal telemetry: Tagging effects, Science (2015).

[14] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Č. Zajc, T. Vojír, G. Häger, A. Lukezic, et al., The Visual Object Tracking VOT2017 Challenge Results, in: Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017, 2018.

[15] F. Maire, L. M. Alvarez, A. Hodgson, A Convolutional Neural Network for Automatic Analysis of Aerial Imagery., DICTA (2014).

[16] F. Romero-Ferrero, M. G. Bergomi, R. Hinz, F. J. H. Heras, G. G. de Polavieja, idtracker.ai: Tracking all individuals in large collectives of unmarked animals, arXiv.org (2018). `arXiv:1803.04351v1`.

[17] D. Held, S. Thrun, S. Savarese, Learning to Track at 100 FPS with Deep Regression Networks., ECCV (2016).

[18] B. Kellenberger, D. Marcos, D. Tuia, Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning, Remote Sensing of Environment (2018).

[19] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: International Conference on Computer Vision, 2011.

[20] M. Brown, D. G. Lowe, Automatic Panoramic Image Stitching using Invariant Features, International Journal of Computer Vision (2006).

[21] Z. M. Bagheri, S. Wiederman, B. Cazzolato, S. Grainger, D. O'Carroll, Performance of an insect-inspired target tracker in natural conditions., Bioinspiration & Biomimetics (2017).

[22] J. M. Graving, D. Chae, H. Naik, L. Li, B. Koger, B. R. Costelloe, I. D. Couzin, DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning, bioRxiv (2019).