

AcinoSet: A 3D Pose Estimation Dataset and Baseline Models for Cheetahs in the Wild

Daniel Joska¹, Liam Clark¹, Naoya Muramatsu², Ricardo Jericevich¹, Fred Nicolls¹, Alexander Mathis³,
Mackenzie W. Mathis³, Amir Patel¹ *Member, IEEE*

Abstract—Animals are capable of extreme agility, yet understanding their complex dynamics, which have ecological, biomechanical and evolutionary implications, remains challenging. Being able to study this incredible agility will be critical for the development of next-generation autonomous legged robots. In particular, the cheetah (*acinonyx jubatus*) is supremely fast and maneuverable, yet quantifying its whole-body 3D kinematic data during locomotion in the wild remains a challenge, even with new deep learning-based methods. In this work we present an extensive dataset of free-running cheetahs in the wild, called *AcinoSet*, that contains 119,490 frames of multi-view synchronized high-speed video footage, camera calibration files and 7,588 human-annotated frames. We utilize markerless animal pose estimation to provide 2D keypoints. Then, we use three methods that serve as strong baselines for 3D pose estimation tool development: traditional sparse bundle adjustment, an Extended Kalman Filter, and a trajectory optimization-based method we call Full Trajectory Estimation. The resulting 3D trajectories, human-checked 3D ground truth, and an interactive tool to inspect the data is also provided. We believe this dataset will be useful for a diverse range of fields such as ecology, neuroscience, robotics, biomechanics as well as computer vision. Code and data can be found at: <https://github.com/African-Robotics-Unit/AcinoSet>.

I. INTRODUCTION

The ability to swiftly and robustly maneuver in the world is paramount to survival for many animals. Leveraging the ability of cheetahs—the fastest land mammal—will be useful to build better legged robots, akin to what has been achieved with flying robots [1]). Maneuverability also presents interesting case studies on how animals have negotiated trade-offs amongst competing requirements such as safety, economy, stability, robustness and agility [2]. The cheetah (*acinonyx jubatus*) is an excellent example of an animal that exhibits complex locomotion. Yet, collecting such complex motion data in laboratory settings, or with traditional motion-capture marker-based systems, is difficult. For instance, GPS-IMU collars treat the animal as a simple point, and harness-based systems are too invasive [3], [4].

Here, we provide a new dataset of cheetahs running “in the wild” named *AcinoSet*, which stems from the scientific name of the cheetah. *AcinoSet* consists high-speed videos taken from six cameras, camera calibration data and over 7,500 hand-labeled 2D key points. Using this dataset we compare three methods of markerless 3D reconstruction: (1)

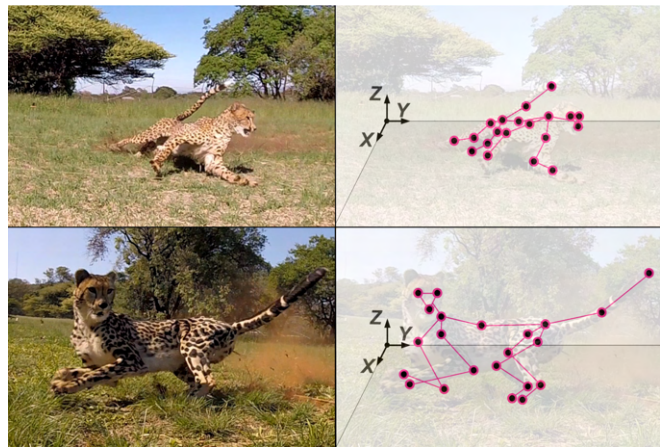


Fig. 1. Example frames from the *AcinoSet* dataset alongside 3D reconstruction. The dataset includes frames from various postures, angles, times of the day (and seasons) from 10 different cheetahs.

multi-view triangulation [5], [6]; (2) an Extended Kalman Filter (EKF) [4]; and (3) a Full Trajectory Estimation (FTE) method, which is inspired by Moving Horizon Estimation typically used in feedback control [7] that outperforms the other methods. We believe this 3D dataset, and our strong baseline FTE method, will serve both the robotic and computer vision community as a new 3D benchmark on this highly articulated animal.

II. RELATED WORK

In recent years several animal datasets have become available [8]. Cao et al. provided around 4k images of several domestic quadrupeds [9]. The MacaquePose dataset provides over 13k annotated images of macaques moving from multiple zoos and other sources [10]. Biggs et al. provide 20K 2D dog postures based on the Stanford Dog Dataset [11]. AnimalWeb provides 21K face images [12]. The Horse10 dataset comprises 8K horses [13]. All these datasets are 2D datasets. OpenMonkeyStudio provided an excellent dataset of 195,228 frames with multiple cameras, but in a “green-studio” laboratory condition [14]. We previously released a set of about 900 Cheetah images in-the-wild [6] but none aimed towards a 3D benchmark animal pose in-the-wild benchmark, as we do in this paper.

Methods for 3D markerless motion capture of humans are benchmarked on large-scale labelled datasets, such as Human 3.6M [15], HumanEva [16], NBA2K [17] or AMASS [18]. Methods (surveyed in [19]) can be divided into model-based [20], [21] and model-free, which in turn can be

Research supported by South African National Research Foundation (Grant No. 117744). ¹ African Robotics Unit (ARU), University of Cape Town, South Africa ² University of Tsukuba, Japan ³ École Polytechnique Fédérale de Lausanne, Switzerland ⁴ Corresponding authors: amir.patel@uct.ac.za, mackenzie.mathis@epfl.ch

divided into lifting [22] and multi-view [23]–[30]. Thereby, the majority of state-of-the-art methods rely on deep learning (see Zheng et al. for an excellent survey [19]). Markerless 3D animal pose estimation has previously been done using photogrammetry [31] but this method requires strong lighting and background texture. An exciting approach is the SMAL method [20], [21], which leverages the 3D scans of toys of animals to obtain 3D shape and texture [11], [32], [33]. Additionally, 2D/3D pose estimation of non-human animals has been used for biomechanical, neuroscience, and other applications [30], [34]–[36]. To advance 3D methods for pose estimation on animals, we provide a benchmark and present three baseline 3D methods.

III. METHODS AND DATA

A. The AcinoSet Dataset

1) **Video Collection:** Footage of 10 cheetahs was captured during enrichment exercises at the Ann van Dyk Centre (Hartbeespoort, South Africa) and Cheetah Outreach (Somerset West, South Africa) in 2017 and 2019. The footage was captured using six GoPro cameras: either a set of Hero 7 Black cameras at a resolution of 2704×1520 at 120 frames per second (fps) or a set of Hero 5 Session cameras at a resolution of 1920×1080 at 90 frames per second (fps), in the configuration depicted in Fig. 2. In total 93 video sequences were collected, consisting of 42 runs (straight galloping) and 51 maneuvers (turning and acceleration). Time synchronization information was included in the fields of view of the cameras by flashing a custom LED rig three times at the beginning and end of each experiment. Calibration (intrinsic and extrinsic) was done using a combination of MATLAB camera calibration toolbox and OpenCV [37] via Sparse Bundle Adjustment [5]. Video collection was approved by the University of Cape Town Science Faculty Animal Ethics Committee.

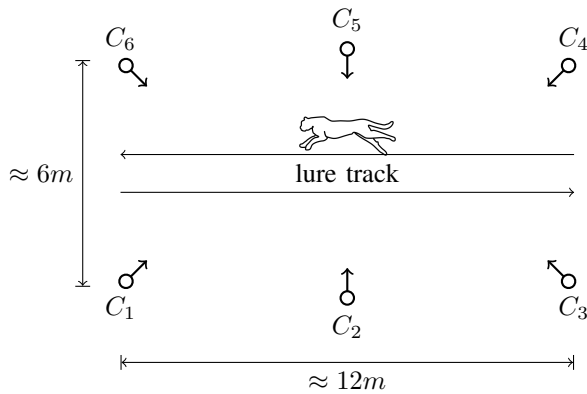


Fig. 2. Camera layout for observing cheetah enrichment exercise runs.

2) **Data Curation:** From these 10 Cheetahs, 7588 images ($\sim 7\%$ of the total video frames), were selected (from a uniform random distribution) and 20 key points were expertly annotated (Fig. 3). Frames from multiple cheetahs (Table I) consisted of a selection of poses within the cheetahs galloping gait as well transient maneuvers (turning, acceleration

and braking, see Fig. 4). Note that the videos have diverse backgrounds, weather, and lighting (Fig. 1).

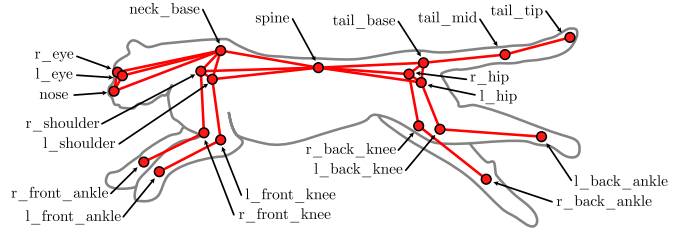


Fig. 3. Cheetah rigid body model with markers locations depicted. The lengths of the links were obtained by a sample of measurements taken from several cheetahs at Ann van Dyk Cheetah Centre (South Africa).

TABLE I
2D DATASET

Cheetah ID	Videos	Days	Total Frames	Human Labeled
Menya	18	2	5,811	853
Jules	117	8	24,853	1450
Phantom	141	8	38,640	1372
Lily	57	4	12,525	1535
Cetane	27	2	8,547	404
Kiara	12	1	2,910	94
Romeo	12	1	2,946	37
Zorro	90	5	21,259	364
BigGirl	8	1	1,396	266
Ebony	3	1	603	155

B. Training 2D Feature Detectors

We utilized an ImageNet pre-trained version of ResNet152, modified within the DeepLabCut 2.0 framework [6], [34], [38]. We also configured the network to use pairwise correspondence [38], which improved the accuracy of the predictions. Data was split into 95% training images, and 5% for testing and trained for $1E6$ iterations with the stochastic gradient descent (SGD) optimizer. The RMSE for test images was 9.09 ± 33.08 pixels. The RMSE for test images for a network trained without pairwise correspondence was 15.5 ± 68.82 pixels. There were still large deviations in RMSE (some with an error of over 2000 pixels) which caused the large standard deviation values. To remove outliers, we fit a Gaussian distribution model. Outliers could roughly be predicted as points deviating by more than 15 pixels (3σ). Then, we obtained a mean of 1.12 pixels and standard deviation of 5.02 pixels. For computing a normalized RMSE (NRMSE) we calculated the bounding box of the cheetah and divided the RMSE by the square root of (height x width).

C. Cheetah Skeletal Model

We present several baseline methods for generating the 3D dataset. For the EKF and the FTE methods, we used a rigid body model of the cheetah (Fig. 3). A set of parameters, shown in Table II, was used to determine the pose of the cheetah model. The model was defined in terms of a global position and orientation of the cheetah and a set of angles describing the relative rotations between the rigid bodies in the model, totalling 24 generalized coordinates.

TABLE II
POSE PARAMETERS FOR THE CHEETAH MODEL

Pose parameters	Description
x, y, z	Head position in inertial frame
ϕ_1, θ_1, ψ_1	Head roll, pitch and yaw in the inertial frame
ϕ_2, θ_2, ψ_2	Neck.base roll, pitch and yaw relative to the head
θ_3	Front torso pitch relative to the neck
θ_4, ψ_4	Back torso pitch and yaw relative to the front torso
θ_5, ψ_5	Tail.base pitch and yaw relative to the back torso
θ_6, ψ_6	Tail.mid pitch and yaw relative to the tail base
θ_7	L.shoulder pitch relative to the front torso
θ_8	L.front.knee pitch relative to the L.shoulder
θ_9	R.shoulder pitch relative to the front torso
θ_{10}	R.front.knee pitch relative to the r.shoulder
θ_{11}	L.hip pitch relative to the back torso
θ_{12}	L.back.knee pitch relative to the L.hip
θ_{13}	R.hip pitch relative to the back torso
θ_{14}	R.back.knee pitch relative to the r.hip

Rotation matrices were defined to relate the orientations of the rigid links in the model; these rotation matrices were defined as coordinate rotations according to [39]. In order to avoid singularities, the order of rotations was chosen such that the second rotation would be less than 90° [39]. Using rigid body kinematics, the 3D positions of the markers could be obtained using the lengths and pose parameters [4].

The positions in the model were mostly based on the predictions from the DeepLabCut model; however, the paws were excluded as the grass occluded them in most of the videos (the ankles are still captured, which allows for a good 3D representation of locomotion). A “head” position, located directly between the eyes of the cheetah, was added to define the cheetah’s position in the inertial frame. The front torso was defined as the rigid body containing the markers neck base, spine and shoulder. Similarly the back torso was comprised of the spine, tail base and hip markers. Measurements of the limbs of a subset of cheetahs were obtained from the Ann van Dyk Cheetah Centre and used to derive the positions as described in Table III. These vary between cheetahs but we found the measurements to be reasonable approximations for the cheetahs.

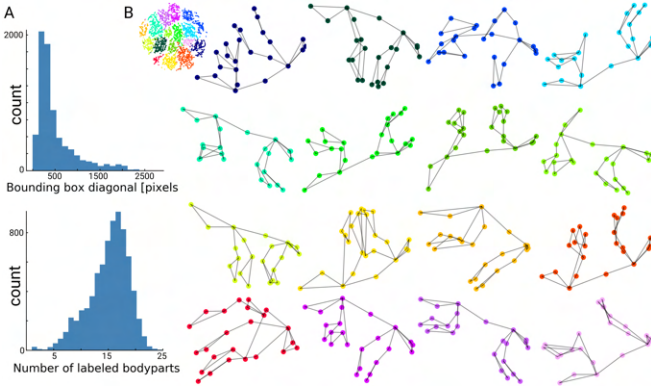


Fig. 4. Dataset statistics A: Histogram of number of key points per annotated image, and range of cheetah sizes. B: Diversity of cheetah poses in annotated data seen by t-SNE clustering of the ground truth data, with example centroid posture from each kmeans cluster (cheetahs normalized for size and clustered by kmeans).

TABLE III
POSITIONS FOR THE KINEMATIC MODEL OF THE CHEETAH

$$\begin{aligned}
\mathbf{p}_{head} &= [x \ y \ z]^T \\
\mathbf{p}_{l.eye} &= \mathbf{p}_{head} + \mathbf{R}_i^1 [0 \ 0.03 \ 0]^T \\
\mathbf{p}_{r.eye} &= \mathbf{p}_{head} + \mathbf{R}_i^1 [0 \ -0.03 \ 0]^T \\
\mathbf{p}_{nose} &= \mathbf{p}_{head} + \mathbf{R}_i^1 [0.055 \ 0 \ -0.055]^T \\
\mathbf{p}_{neck.base} &= \mathbf{p}_{head} + \mathbf{R}_i^2 [-0.28 \ 0 \ 0]^T \\
\mathbf{p}_{spine} &= \mathbf{p}_{neck.base} + \mathbf{R}_i^3 [-0.37 \ 0 \ 0]^T \\
\mathbf{p}_{tail.base} &= \mathbf{p}_{spine} + \mathbf{R}_i^4 [-0.37 \ 0 \ 0]^T \\
\mathbf{p}_{tail.mid} &= \mathbf{p}_{tail.base} + \mathbf{R}_i^5 [-0.28 \ 0 \ 0]^T \\
\mathbf{p}_{tail.tip} &= \mathbf{p}_{tail.mid} + \mathbf{R}_i^6 [-0.36 \ 0 \ 0]^T \\
\mathbf{p}_{l.shoulder} &= \mathbf{p}_{neck.base} + \mathbf{R}_i^3 [-0.04 \ 0.08 \ -0.10]^T \\
\mathbf{p}_{l.front.knee} &= \mathbf{p}_{l.shoulder} + \mathbf{R}_i^7 [0 \ 0 \ -0.24]^T \\
\mathbf{p}_{l.front.ankle} &= \mathbf{p}_{l.front.knee} + \mathbf{R}_i^8 [0 \ 0 \ -0.28]^T \\
\mathbf{p}_{r.shoulder} &= \mathbf{p}_{neck.base} + \mathbf{R}_i^3 [-0.04 \ -0.08 \ -0.10]^T \\
\mathbf{p}_{r.front.knee} &= \mathbf{p}_{r.shoulder} + \mathbf{R}_i^9 [0 \ 0 \ -0.24]^T \\
\mathbf{p}_{r.front.ankle} &= \mathbf{p}_{r.front.knee} + \mathbf{R}_i^{10} [0 \ 0 \ -0.28]^T \\
\mathbf{p}_{l.hip} &= \mathbf{p}_{tail.base} + \mathbf{R}_i^4 [0.12 \ 0.08 \ -0.06]^T \\
\mathbf{p}_{l.back.knee} &= \mathbf{p}_{l.hip} + \mathbf{R}_i^{11} [0 \ 0 \ -0.32]^T \\
\mathbf{p}_{l.back.ankle} &= \mathbf{p}_{l.back.knee} + \mathbf{R}_i^{12} [0 \ 0 \ -0.25]^T \\
\mathbf{p}_{r.hip} &= \mathbf{p}_{tail.base} + \mathbf{R}_i^4 [0.12 \ 0.08 \ -0.06]^T \\
\mathbf{p}_{r.back.knee} &= \mathbf{p}_{r.hip} + \mathbf{R}_i^{13} [0 \ 0 \ -0.32]^T \\
\mathbf{p}_{r.back.ankle} &= \mathbf{p}_{r.back.knee} + \mathbf{R}_i^{14} [0 \ 0 \ -0.25]^T
\end{aligned}$$

IV. RESULTS

The main contributions of this work are three-fold:

- 1) The large 2D dataset ($>100K$ frames, with 7588 human annotated frames, as described in Methods & Data).
- 2) Three baseline methods for generated the 3D data, including FTE, which provides excellent results.
- 3) Human-validated 3D ground-truth frames and graphical user interface tools to validate more.

To create a strong baseline 3D dataset, we used three methods (Fig. 5) to compute 3D skeletons, which utilized the 2D key point estimates from the 119,490 video frames.

A. Triangulation

Our first baseline is sparse bundle adjustment [5] using SciPy [40]. Unlike in our prior work [6], we used a Cauchy robust cost function to prevent outliers from skewing the 3D point estimates to the naive squared error approach [5]. The initial 3D point estimates were obtained for 2D correspondences with a likelihood above 0.5.

B. Extended Kalman Filter

Our second baseline is another popular method for multi-camera 3D reconstruction, namely the Extended Kalman Filter (EKF) [41]. In our configuration the EKF utilizes the rigid body model described in Section III-C and factors in the covariances (measurement and model) to increase the accuracy of the state estimates.

1) *Process Model*: The skeletal motion was assumed to behave according to a constant acceleration model (with sample time Δt). The kinematic state vector $\hat{\mathbf{x}}$ therefore consisted of the pose parameters \mathbf{q} (Table II), their velocities $\dot{\mathbf{q}}$ and their accelerations $\ddot{\mathbf{q}}$. Jerk in the pose parameters i was

accounted for as a tunable covariance (σ_i) for smoothness [42].

2) *Measurement Model*: A measurement function for the EKF was required to produce a set of measurement estimates given a set of state estimates $\hat{\mathbf{x}}$. The 3D positions of each marker for the cheetah could be determined using the 3D kinematic equations. These 3D positions were then projected into the view of each camera taking into account the fisheye distortion effects [4]. Denoting the estimated measurement for coordinate c (either u or v) for a marker m seen by camera i as $\hat{\mathbf{z}}_{i,m,c}$, the measurement equation provides a measurement estimate $\hat{\mathbf{z}}$ comprising 240 elements (6 cameras \times 20 markers \times 2 pixel coordinates):

$$\hat{\mathbf{z}} = h(\hat{\mathbf{x}}) = \begin{bmatrix} \hat{\mathbf{z}}_{1,eye,u} \\ \hat{\mathbf{z}}_{1,eye,v} \\ \vdots \\ \hat{\mathbf{z}}_{1,r.back.ankle,v} \\ \hat{\mathbf{z}}_{2,eye,u} \\ \vdots \\ \hat{\mathbf{z}}_{6,r.back.ankle,v} \end{bmatrix}. \quad (1)$$

The measurement covariance matrix \mathbf{R} was constructed as a diagonal matrix where each diagonal element was equal to the measurement variance from DeepLabCut ($\sigma^2 = (5 \text{ pixels})^2$) for high likelihoods and 2704^2 for low likelihoods, which is the square of the maximum possible pixel error when a cheetah was in the frame: $\mathbf{R} = I\sigma^2$.

3) *Outlier Rejection*: In order to prevent outliers from causing the state to diverge, an extra step was added to the update stage of the EKF: if an outlier was detected, the innovation (residual) for that measurement was set to zero. An outlier could be determined using the innovation, $\tilde{\mathbf{y}}$, and innovation covariance, \mathbf{S} . Outliers were assumed to be measurements whose innovation was three times greater than the square root of the innovation covariance [43].

C. Full Trajectory Estimation

One downside of the EKF is that estimates only depend on the previous sample and can diverge if multiple outliers are present in time. Thus, we consider the entire trajectory of states and measurements simultaneously. This also allows us to impose state constraints which is not possible in the EKF, akin to Moving Horizon Estimation (MHE) [7], yet here we optimize over the full trajectory.

1) *Parameters and variables*: A trajectory, \mathbf{x} , consists of a set of pose parameters at each time step. These pose parameters are the positions and angles, $[x, y, z, \phi_1 \dots \theta_{14}]$, which allow us to define the 3D marker positions of the cheetah. The first and second order time derivatives of \mathbf{x} , namely $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$, describe the “velocities” and “accelerations” of the pose parameters respectively. An array, \mathbf{s} , contained the 3D marker positions for each time step. The complete list of variables are listed in Table IV. An array, \mathbf{y} , contained 2D keypoints with shape $N \times c \times m \times 2$, with N the number of time steps in the trajectory, p the number of pose parameters (24), c the number of cameras (6) and m the number of markers on the cheetah (20).

TABLE IV
TRAJECTORY OPTIMIZATION VARIABLES.

Variables	Length	Description
\mathbf{x}	$N \times p$	Pose parameters across trajectory
$\dot{\mathbf{x}}$	$N \times p$	Time derivative of \mathbf{x}
$\ddot{\mathbf{x}}$	$N \times p$	Time derivative of $\dot{\mathbf{x}}$
\mathbf{s}	$N \times m \times 3$	3D marker positions
\mathbf{w}	$N \times p - 1$	Model noise
\mathbf{v}	$N \times c \times m \times 2$	Measurement noise

2) *Modeling Constraints*: Similar to the EKF, the model constraints between time steps used a constant acceleration model. In reality, the pose parameters did not obey this model and so \mathbf{w} was added to account for the acceleration errors (or *disturbances*) for each pose parameter between time steps. Implicit Euler integration was used to formulate the equations below:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta t \dot{\mathbf{x}}_k \quad \text{for } k = \{2, \dots, N\} \quad (2)$$

$$\dot{\mathbf{x}}_k = \dot{\mathbf{x}}_{k-1} + \Delta t \ddot{\mathbf{x}}_k \quad \text{for } k = \{2, \dots, N\} \quad (3)$$

$$\ddot{\mathbf{x}}_k = \ddot{\mathbf{x}}_{k-1} + \mathbf{w}_k \quad \text{for } k = \{2, \dots, N\}. \quad (4)$$

In addition to the constraints for the motion model, bound constraints on the pose parameters were added to ensure physically plausible motion.

3) *Measurement constraints*: The estimated pose parameters need to be related to the measurements in order to determine how well the estimated pose relates to what is seen by the cameras. In order to simplify the measurement estimate constraints, the problem was split into two parts. First, a constraint was set up that relates the pose parameters at each time step to a set of 3D marker positions through a function, f :

$$\mathbf{s}_k = f(\mathbf{x}_k) \quad \text{for } k = \{2, \dots, N\}. \quad (5)$$

A second constraint was set up which projected these 3D marker positions to 2D measurement estimates, $\hat{\mathbf{y}}$, for each camera and equated the measurement estimates to the measurements, \mathbf{y} , from DeepLabCut. A variable \mathbf{v} was added to account for measurement noise or *measurement disturbances*. A projection function, h , for each camera was used to project the 3D points in order to obtain the 2D measurement estimates:

$$\mathbf{y}_{i,j} = h_j(\mathbf{s}_{i,j}) + \mathbf{v}_{i,j} \quad \text{for } i = \{1, \dots, n\}, \quad j = \{1, \dots, c\}. \quad (6)$$

4) *Cost function*: Given the constraints above, we have a set of measurement errors, \mathbf{v} , and acceleration errors, \mathbf{w} , which we would like to minimize. The measurement error, e_{meas} , can be formulated in a way that is very similar to that of the triangulation. In order to reduce the effect of outliers, the re-descending robust cost function $C(\cdot)$ [44] with values $a = 3, b = 10, c = 20$ was used.

The measurement costs were normalised by dividing by the standard deviation of the measurements, $\sigma_{meas} = 5$ pixels:

$$e_{meas} = \sum_{i=1}^n \sum_{j=1}^c \sum_{k=1}^m \sum_{l=1}^2 C\left(\frac{\mathbf{v}_{i,j,k,l}}{\sigma_{meas}}\right). \quad (7)$$

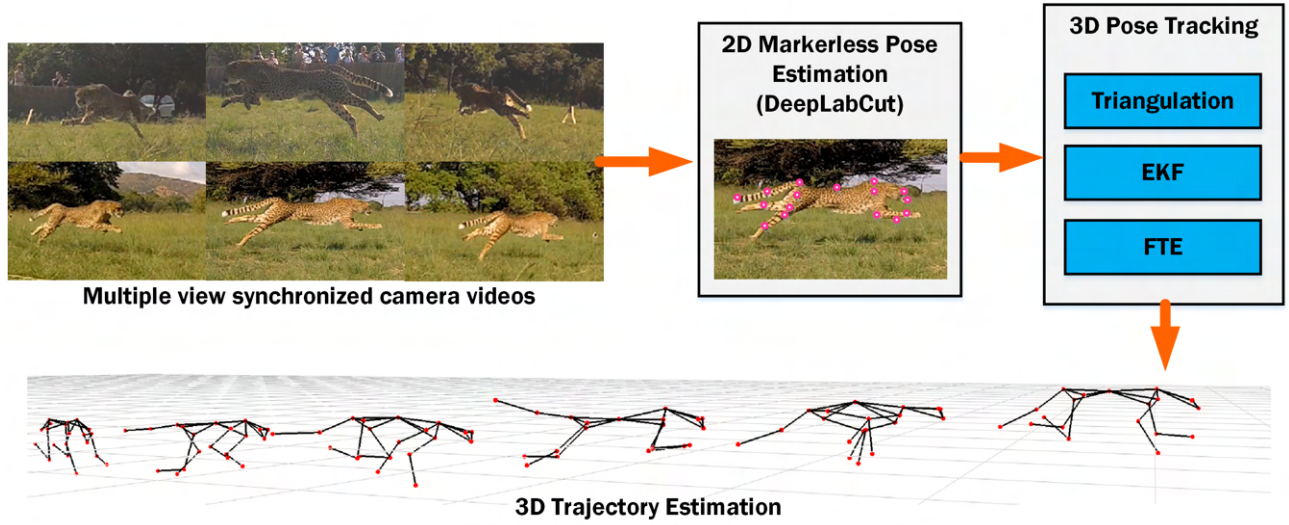


Fig. 5. **An overview of the 3D pose tracking.** Multi-view videos are fed into a markerless pose estimation algorithm (DeepLabCut) which outputs 2D pose estimates in each view. These views are then combined using one of the three 3D pose estimation methods (Triangulation, EKF or FTE).

The model error, e_{model} , was the sum of squares of acceleration noise. In order to normalise this noise, the squared acceleration error values were divided by the acceleration variance of the pose parameter they correspond to. Alternatively, the normalisation can be thought of as dividing by the standard deviation before squaring the value:

$$e_{model} = \sum_{i=1}^n \sum_{j=1}^p \frac{\mathbf{w}_{i,j}^2}{\sigma_{modelj}^2} = \sum_{i=1}^n \sum_{j=1}^p \left(\frac{\mathbf{w}_{i,j}}{\sigma_{modelj}} \right)^2. \quad (8)$$

The minimization problem could then simply be formulated as per (9) subject to all of the model and measurement constraints:

$$\min_{\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \mathbf{s}, \mathbf{w}, \mathbf{v}} e_{meas} + e_{model}. \quad (9)$$

5) *Implementation:* The optimization was implemented using Pyomo [45] and IPOPT [46] with the MA86 linear solver [47]. Similarly to the EKF, for each run the position and orientation of the cheetah was estimated using the triangulated points. All of the other states were initialised to zero (so that the cheetah was in a valid, upright pose).

V. 3D BASELINE MODEL TESTING

First, to evaluate each of the 3D reconstruction techniques, we constructed synthetic ground truth data (to mitigate any errors in feature detection, camera calibration, or human-annotation). Thus, a cheetah run was simulated by defining a 3D pose (using the intrinsic and extrinsic parameters we measured), corresponding to marker positions, for a cheetah as a function of pose parameters. A 3D trajectory was then created by adjusting the pose parameters at each frame to create a realistic-looking cheetah run.

Several simulated datasets were used for testing and evaluating the different trajectory estimation methods; specifically, datasets which allowed for the evaluation of these methods in the presence of noise and outliers in the measurement data (the 2D points seen by the cameras). Varying degrees

of Gaussian noise were added to the ground truth 2D points. The noise n for each 2D coordinate was randomly drawn from a Gaussian distribution with 0 mean and standard deviation σ_n , so $n \sim \mathcal{N}(0, \sigma_n^2)$.

Outliers were created by adding an “outlier value” o to a randomly selected subset of the ground truth 2D points. The probability of a 2D point being turned into an outlier was determined by p_o . The outlier value o for each 2D coordinate was randomly drawn from a Gaussian distribution with 0 mean and standard deviation σ_o , so $o \sim \mathcal{N}(0, \sigma_o^2)$.

Each modified simulated 2D point coordinate c_{noisy} was obtained using

$$c_{noisy} = \begin{cases} c + n & \text{for noise only} \\ c + n + o & \text{for outliers (including noise)} \end{cases} \quad (10)$$

where c is one of the (u, v) pixel coordinates.

The results for all three methods are summarised in Fig. 6, which compares the median absolute difference and median.

A. Baseline 3D on AcinoSet

All three methods were next validated on the *AcinoSet* (Fig. 7). A visual comparison of the three methods can also be found in the supplementary video.

To quantitatively evaluate each of the three methods, we reprojected 3D point predictions for the different methods to the camera planes of each of the 6 cameras. Each of the methods was given filtered multi-view 2D data (points with a confidence value of less than 0.5 were discarded, as they are often occluded). We then compared these reprojected 2D points with 2D ground truth data for the corresponding frames using the RMSE. Two poses were considered: one simple “run” pose with high visibility for all body parts and minimal occlusion, and one more complicated “dive” pose with low visibility and high levels of occlusion (see Fig. 7). For each pose, 1000 hand-checked 3D ground truth points were reprojected to each of the 6 image planes, for a total of

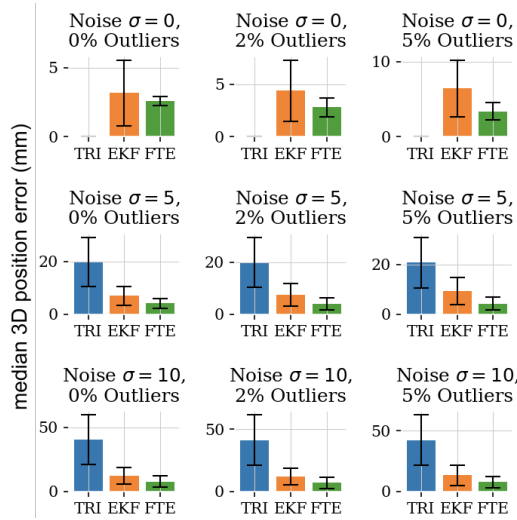


Fig. 6. Summary of simulation results of the three methods.

6000 2D ground truth points. The RMSE and standard error of mean (SEM), along with a normalised error (NRMSE) scaling with the size of the cheetah in frame, for each method are shown in Table VII. Consistent with the visual impression from Fig. 7, full trajectory estimation performed best.

TABLE V
RMSE, SEM, AND NRMSE IN PIXELS FOR EACH METHOD

Pose Type	Metric	TRI	EKF	FTE
Run	RMSE	28.24	3.40	2.76
	SEM	0.26	0.03	0.03
	NRMSE	0.17	0.02	0.02
Dive	RMSE	76.35	39.40	38.44
	SEM	0.70	0.36	0.35
	NRMSE	0.56	0.29	0.28

VI. 3D GROUND TRUTH VALIDATION

As the FTE performed the best, we processed all the >119K frames to create the *3D AcinoSet*. This results in 19,915 3D frames. To check the quality of these 3D postures—as this dataset should serve as a strong benchmark for 3D pose estimation algorithm development—we built a new graphical user interface in order to swiftly check, or correct, 3D ground truth as needed. We used this tool to estimate the amount of human corrections needed to be “perfect” (most needed no corrections, and minor corrections were under 10 pixels, checked on $n=600$ frames) and the largest adjustments were typically in the extremities, such as the tail and ankles. The proportion of large outliers where the adjustment was over 100 mm in magnitude was only 0.71%.

VII. DISCUSSION AND CONCLUSIONS

We present the development of a new animal multi-camera pose estimation dataset, termed *AcinoSet*, for 3D pose estimation. With over 7500 human-annotated images, plus 6 camera data and calibration information, we believe this dataset will be an attractive platform for benchmarking new 3D pose estimation tools. This dataset may be beneficial for researchers in fields such as biology and robotics: in biology,

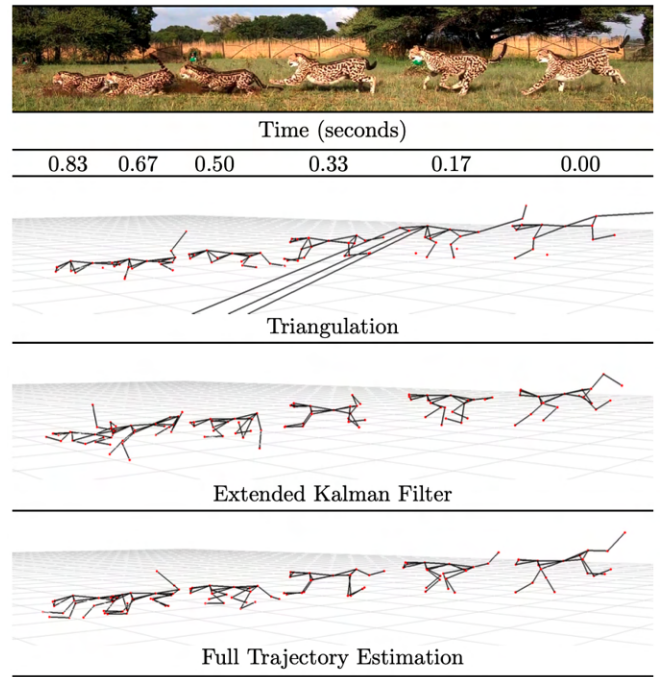


Fig. 7. Examples of the 3D baseline models run on real cheetah footage.

there is a push to bring biomechanics into the field [48], [49], and in robotics, for researchers developing deep-learning based controllers [50].

We also compared three baseline methods for 3D reconstruction. We report that the most robust method, both quantitatively and qualitatively, was the FTE method. Unsurprisingly, this and the EKF outperformed triangulation, likely as they also model the temporal continuity at the cost of greater computational complexity.

The two main advantages of FTE are that the pose parameters can be constrained to their natural ranges, and the 3D marker positions constrained to the skeletal model of the cheetah. It does however require a good initial seed, perhaps provided by either the triangulation or the EKF as a warm-start. The EKF method provided a large improvement over the triangulation with the inclusion of temporal information (via states) and a skeletal model. The rigid body model applied is by no means a realistic skeletal model of the cheetah, yet it provides enough structure to generate acceptable 3D trajectories. Taken together, we hope this allows the community to tackle new 3D challenges.

ACKNOWLEDGMENTS

The authors thank An Chi Chen, Alexandra Barry, Bilal Waleed, Inessa Rajah, James Cushway & Annet George for assistance with video collection, labelling & NN training. We also thank Ines Everaert of Ann van Dyk Centre & Liesl Smith of Cheetah Outreach for allowing access to their cheetahs. MWM is the Bertarelli Foundation Chair of Integrative Neuroscience.

REFERENCES

- [1] N. T. Jafferis, E. F. Helbling, M. Karpelson, and R. J. Wood, "Untethered flight of an insect-sized flapping-wing microscale aerial vehicle," *Nature*, vol. 570, no. 7762, pp. 491–495, 2019.
- [2] M. A. Daley and J. Bertram, "Non-steady locomotion," *Understanding mammalian locomotion: concepts and applications*, pp. 277–306, 2016.
- [3] A. M. Wilson, J. Lowe, K. Roskilly, P. E. Hudson, K. Golabek, and J. McNutt, "Locomotion dynamics of hunting in wild cheetahs," *Nature*, vol. 498, no. 7453, pp. 185–189, 2013.
- [4] A. Patel, B. Stocks, C. Fisher, F. Nicolls, and E. Boje, "Tracking the cheetah tail using animal-borne cameras, gps, and an imu," *IEEE sensors letters*, vol. 1, no. 4, pp. 1–4, 2017.
- [5] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [6] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis, "Using deeplabcut for 3d markerless pose estimation across species and behaviors," *Nature protocols*, vol. 14, no. 7, pp. 2152–2176, 2019.
- [7] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE transactions on automatic control*, vol. 48, no. 2, pp. 246–258, 2003.
- [8] A. Mathis, S. Schneider, J. Lauer, and M. Mathis, "A primer on motion capture with deep learning: Principles, pitfalls, and perspectives," *Neuron*, vol. 108, pp. 44–65, 2020.
- [9] J. Cao, H. Tang, H.-S. Fang, X. Shen, C. Lu, and Y.-W. Tai, "Cross-domain adaptation for animal pose estimation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [10] R. Labuguen, J. Matsumoto, S. Negrete, H. Nishimaru, H. Nishijo, M. Takada, Y. Go, K.-i. Inoue, and T. Shibata, "Macaquepose: A novel 'in the wild' macaque monkey pose dataset for markerless motion capture," *bioRxiv*, 2020.
- [11] B. Biggs, O. Boyne, J. Charles, A. Fitzgibbon, and R. Cipolla, "Who left the dogs out? 3d animal reconstruction with expectation maximization in the loop," in *European Conference on Computer Vision*, pp. 195–211, Springer, 2020.
- [12] M. H. Khan, J. McDonagh, S. Khan, M. Shahabuddin, A. Arora, F. S. Khan, L. Shao, and G. Tzimiropoulos, "Animalweb: A large-scale hierarchical dataset of annotated animal faces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6939–6948, 2020.
- [13] A. Mathis, T. Biasi, S. Schneider, M. Yuksekgonul, B. Rogers, M. Bethge, and M. W. Mathis, "Pretraining boosts out-of-domain robustness for pose estimation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1859–1868, 2021.
- [14] P. C. Bala, B. R. Eisenreich, S. B. M. Yoo, B. Y. Hayden, H. S. Park, and J. Zimmermann, "Automated markerless pose estimation in freely moving macaques with openmonkeystudio," *Nature communications*, vol. 11, no. 1, pp. 1–12, 2020.
- [15] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.
- [16] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1–2, p. 4, 2010.
- [17] L. Zhu, K. Rematas, B. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Reconstructing nba players," in *European Conference on Computer Vision*, pp. 177–194, Springer, 2020.
- [18] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5442–5451, 2019.
- [19] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.
- [20] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.
- [21] Y. Huang, F. Bogo, C. Lassner, A. Kanazawa, P. V. Gehler, J. Romero, I. Akhter, and M. J. Black, "Towards accurate marker-less human shape and pose estimation over time," in *2017 international conference on 3D vision (3DV)*, pp. 421–430, IEEE, 2017.
- [22] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnct: Real-time multi-person 3d motion capture with a single rgb camera," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 82–1, 2020.
- [23] M. Burenus, J. Sullivan, and S. Carlsson, "3d pictorial structures for multiple view articulated pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3618–3625, 2013.
- [24] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3d pictorial structures for multiple human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1669–1676, 2014.
- [25] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation for 3d human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 750–767, 2018.
- [26] A. Arnab, C. Doersch, and A. Zisserman, "Exploiting temporal context for 3d human pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3395–3404, 2019.
- [27] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, "Cross view fusion for 3d human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4342–4351, 2019.
- [28] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5255–5264, 2018.
- [29] C.-H. Chen, A. Tyagi, A. Agrawal, D. Drover, S. Stojanov, and J. M. Rehg, "Unsupervised 3d pose estimation with geometric self-supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5714–5724, 2019.
- [30] Y. Yao, Y. Jafarian, and H. S. Park, "Monet: Multiview semi-supervised keypoint detection via epipolar divergence," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 753–762, 2019.
- [31] W. I. Sellers and E. Hirasaki, "Markerless 3d motion capture for animal locomotion studies," *Biology open*, vol. 3, no. 7, pp. 656–668, 2014.
- [32] S. Zuffi, A. Kanazawa, and M. J. Black, "Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3955–3963, 2018.
- [33] S. Zuffi, A. Kanazawa, T. Berger-Wolf, and M. Black, "Three-d safari: Learning to estimate zebra pose, shape, and texture from images 'in the wild'," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5358–5367, IEEE, 2019.
- [34] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "Deeplabcut: markerless pose estimation of user-defined body parts with deep learning," *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.
- [35] B. Biggs, T. Roddick, A. Fitzgibbon, and R. Cipolla, "Creatures great and SMAL: Recovering the shape and motion of animals from video," in *ACCV*, 2018.
- [36] J. M. Graving, D. Chae, H. Naik, L. Li, B. Koger, B. R. Costelloe, and I. D. Couzin, "Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning," *Elife*, vol. 8, p. e47994, 2019.
- [37] G. Bradski, "The OpenCV Library," *Dr. Dobbs's Journal of Software Tools*, 2000.
- [38] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepcruc: A deeper, stronger, and faster multi-person pose estimation model," in *European Conference on Computer Vision*, pp. 34–50, Springer, 2016.
- [39] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15–16, pp. 1–35, 2006.
- [40] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.
- [41] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.
- [42] R. Labbe, "Kalman and bayesian filters in python." <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python>, 2014.

- [43] H. Fang, M. A. Haile, and Y. Wang, "Robustifying the kalman filter against measurement outliers: An innovation saturation mechanism," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6390–6395, IEEE, 2018.
- [44] B. Nicholson, R. López-Negrete, and L. T. Biegler, "On-line state estimation of nonlinear dynamic systems with gross errors," *Computers & chemical engineering*, vol. 70, pp. 149–159, 2014.
- [45] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Sirola, *Pyomo—optimization modeling in python*, vol. 67. Springer Science & Business Media, second ed., 2017.
- [46] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [47] A. HSL, "collection of fortran codes for large-scale scientific computation," See <http://www.hsl.rl.ac.uk>, 2007.
- [48] U. Bauer, S. Poppinga, and U. K. Müller, "Mechanical ecology—taking biomechanics to the field," *Integrative and Comparative Biology*, 2020.
- [49] S. B. Hausmann, A. M. Vargas, A. Mathis, and M. W. Mathis, "Measuring and modeling the motor system with machine learning," *arXiv preprint arXiv:2103.11775*, 2021.
- [50] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.