2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Autonomous Rotating Cameras Boost 3D Wildlife Motion Capture Yield without Human Operators

Amaan Vally[1], Daniel Joska[1], Naoya Muramatsu[1],
Paul Amayo[1] *Member, IEEE*, Amir Patel[1,2†] *Senior Member, IEEE*

***Abstract*— We present a low-cost, autonomous, rotating-camera system that increases the usable data yield for 3D markerless motion capture of animals in uncontrolled outdoor settings. A lightweight detector (YOLOv4-Tiny) locates the subject at 10 Hz; an Extended Kalman Filter (EKF) bridges sparse detections to a 50 Hz full-state feedback (FSF) controller, keeping the subject centered without a human operator. The 3D reconstruction backend uses existing markerless 2D keypoints and Full Trajectory Estimation (FTE) with a simple rotation compensation for moving cameras. On field videos of a running human and free-running cheetahs, the rotating cameras captured substantially more usable frames than fixed cameras: +52% for the human sequence (6593 vs. 4333 frames) and +135% across cheetah sequences (2419 vs. 1031 frames). Centering also shifts the subject-pixel distribution toward the image center, which theoretically lowers 2D keypoint error and thus 3D reprojection error for any pose-estimation backend. We detail the EKF design for sparse/noisy detections, the FSF controller with an integral state, and practical deployment considerations. Results show autonomous centering is a simple, deployable lever to scale outdoor animal motion capture without changing downstream reconstruction methods.**

markerless motion capture, rotating cameras, subject tracking

I. Introduction

Animal motion capture is an active field of research within robotics and computer vision and has a variety of applications, such as in biomechanics and biomimetics [1]. Motion capture research is useful for understanding the kinematics of complex organic bodies in motion, and we can use this understanding to design better performing robots by mimicking what we see in nature [2]. 3D markerless motion capture is the gold standard but has been limited by the complexity of accurately acquiring the 3D kinematics of an object in motion without the spatial and temporal correspondence that can be relied on in marker-based approaches [3]. A common pitfall with 3D markerless motion capture systems that use RGB sensors (cameras) is the amount of usable data that can be collected in the wild due to the sensor's limited field of view. Approaches to combat this require a human operator, or the use of larger camera setups or autonomous drones, making it prohibitively costly or complex and overall unsuitable for 3D markerless motion capture of animals in the wild [4]–[7]. This work contributes:

*This research was supported by South African National Research Foundation (Grant Nos. 137762, 131705 and 132751).

[1] African Robotics Unit (ARU), University of Cape Town, South Africa
[2] Department of Computer Science, University College London, UK
†Corresponding email: amir.patel@ucl.ac.uk

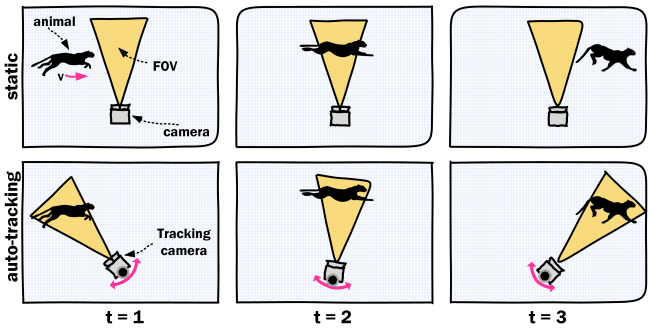

Fig. 1. Figure showing how the tracking system allows us to increase the amount of data collected by automatically rotating to keep a fast-moving target in view.

- **An autonomous, rotating camera system** that actively centers uncooperative outdoor subjects, requiring no operator and working with commodity sensors.
- **A temporal-bridging pipeline** ($10\,\mathbf{Hz} \rightarrow 50\,\mathbf{Hz}$): an EKF with a constant-acceleration model and a noisy pseudo-measurement to remain stable through detector dropouts, feeding an FSF position controller with an integral state.
- **A moving-camera 3D reconstruction recipe**: plug-and-play with existing 2D keypoints and FTE by applying a simple yaw rotation from the encoder to recover world-frame trajectories.
- **Evidence of yield improvement**: substantially more frames with the subject in-view and more time spent near the image center for both a running human and free-running cheetahs, using only field videos (no special markers/templates).

In this paper, we present a new camera system which can autonomously track and reconstruct the motion of a moving animal in the wild. The mechatronic design of the system is discussed in Section III.A. Section III.B explains how we utilize an existing object detection neural network to locate the subject in the video stream from a webcam mounted to a motor, and combine an EKF and a FSF controller to control motor position and keep the subject centered in the frame. Section III.C describes the use of FTE for reconstruction of 3D trajectories. Finally, in Sections IV and V we present our results and discussion thereof.

II. Related Work

We group prior work into three categories: marker-based animal motion capture, fixed-camera markerless motion

capture, and moving-camera or actively tracked systems. We briefly review each category below and position our system as an upstream field-instrumentation improvement that increases usable capture yield for existing 2D-to-3D reconstruction pipelines.

A. 3D Marker-based Pose Estimation

Motion capture of humans has been a topic of research for decades and is already widely and successfully used in many industries, but far less work has been done on motion capture of wild animals in uncontrolled environments. This is partially because motion capture techniques historically relied on marker-based approaches, which require extensive preparation and a controlled environment. They also require that the subject wear motion capture suits with reflective markers [8], [9] or sensors such as IMUs [10], [11] or pressure sensors [12], rendering these techniques unsuitable for motion capture of animals in the wild.

Marker-based approaches for 3D pose estimation in animals have so far been demonstrated mainly on domesticated species. For example, [13] uses a commercially available Vicon system [8] to capture and compare the motion of Doberman Pinschers with and without Cervical Spondylomyelopathy, while [14] uses another commercially available system to observe the strategies that cats use for obstacle avoidance during walking.

Although marker-based motion capture can be highly accurate, the use of markers has several limitations: (i) Markers attached to the subject can affect the subject's movement, (ii) The time required for marker placement can be excessive. (iii) The markers on the subject's skin can move relative to the underlying bone leading to errors. More importantly, the use of marker-based techniques makes it nearly impossible to perform motion capture of subjects "in the wild", as special preparation is needed [3].

B. 3D Markerless Pose Estimation

Recent methods for human markerless motion capture (see the survey by [15]) are either model-based [16] [17] or model-free, and can be further decomposed into lifting [18] or multi-view [19] - [26]. The majority of state-of-the-art methods rely on deep-learning [15].

Most markerless motion capture systems use single- or multi-view videos from RGB sensors. Open source software packages such as *Argus* [27] allow objects or points to be manually labelled and tracked across frames in 2D videos from calibrated cameras, from which 3D points can be estimated using geometric techniques. Labelling 2D points manually is a time-consuming process, but the recent development of software that uses machine-learning to do this has made the process far easier. Recent work by [28], [29] provide software packages that allow the user to label a portion of their data-set, which is then used to train a neural network that does the rest of the labelling once trained. DeepLabCut (DLC) is the most widely used and also allows users to triangulate 3D points from two static cameras [3] with [30] extending the work to provide support for (static) multi-camera setups. [31] reviews the principles of video (and stereo video) analysis in 3D markerless motion capture, and also provides open source software. Photogrammetric approaches to 3D markerless pose estimation for animals do exist [32], but these methods are dependent on background texture and good lighting. SMAL is another exciting method [16] [17], which obtains 3D shape and texture from 3D scans of toy animals [33] [34] [35].

Other work in 3D markerless pose estimation for animals focus on specific species such as monkeys [36], [37], cheetahs [38], dogs [39] and octopuses [40]. Works by [41] and [42] present data-sets for machine-learning based 3D markerless motion capture consisting of images with labelled key-points of free-running cheetahs and macaque monkeys in naturalistic settings. The majority of these methods and data-sets have been developed for and rely on static cameras and have an inherently restrictive capture area. Our work aims to increase the amount of usable data that can be collected by making use of a rotating camera system to achieve a much larger capture area.

C. Moving Cameras for Markerless Motion Capture

Prior moving-camera systems either require controlled template scans or human camera operators (e.g., hand-held/sparse rigs, on-set capture, outdoor markerless with pre-scanned subjects), which limits use with wild animals [43], [44]. UAV-based systems such as FlyCap further require a template scanning stage with cooperative subjects and RGB-D hardware [6]. Other wildlife trackers treat animals as points and therefore cannot recover full body kinematics [45], [46]. In contrast, our system autonomously keeps uncooperative subjects centered using commodity RGB cameras, integrates cleanly with existing 2D-to-3D pipelines, and empirically increases the number of usable frames captured in the field.

III. Methods and Materials

This section begins by first describing the mechatronic design of our system in Section III.A. Following this, in Section III.B we describe the EKF and controller which the tracking system is comprised of. Finally, in Section III.C we discuss the implementation of FTE with rotational compensation to perform 3D markerless pose estimation.

A. Mechatronic Design

To actuate the rotating sensor pedestal, we utilise a Maxon 18V 70W brush-less DC motor with 500-count quadrature encoder and a gearbox with 51:1 ratio. We also use a Maxon 50/5 servo controller. A Teensy 4.0 micro-controller was used for hardware interfacing via pulse-width modulation (PWM). A Jetson Nano 4GB single-board computer is used for the interfacing with the tracking camera (Logitech C920 HD USB Webcam) and performing inference/object detection. These are all mounted onto a 300x300x300mm frame made of PG20 aluminium extrusion, with 5mm thick aluminium sheet attached to the top and bottom. A sensor pedestal is mounted to the motor shaft for attaching the tracking camera

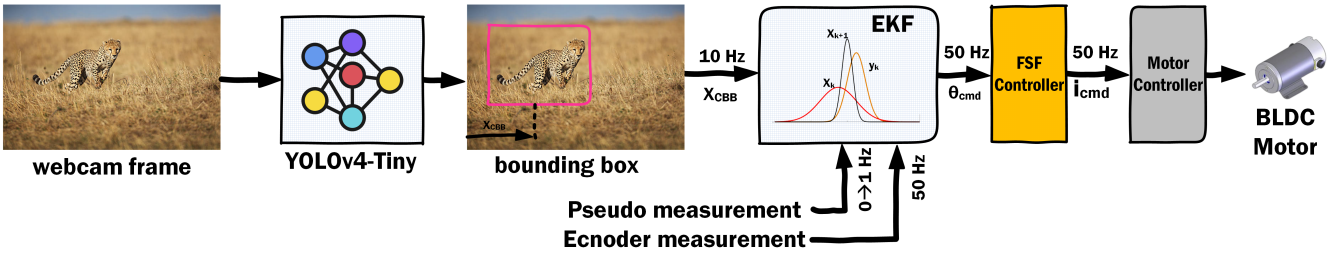

Fig. 2. Basic flow of operation and the data transmission rates for the tracking system are shown.

and other sensors. A 3D render of the complete system is shown in Fig. 3. In a typical deployment, we place the rig near the expected motion corridor, calibrate the cameras, home the rotary encoder to define the yaw reference, start the detector and controller, and then record synchronized static and rotating video streams.

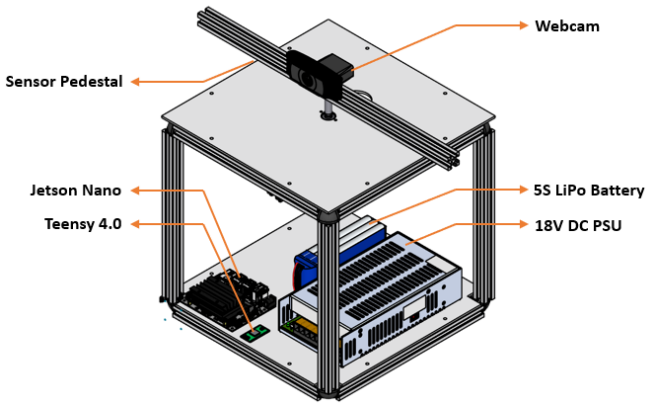

Fig. 3. 3D Render of the Tracking system showing the key components. Sensors such as cameras, LIDAR, or mmWave radars can be attached to the sensor pedestal.

B. Subject Tracking

1) Object Detection: The subject's location relative to the system needs to be determined before the subject can be tracked. To do this, we use a popular object detection convolutional neural network, YOLOv4, which offers a favorable trade-off between accuracy and speed for our application [47]. Due to computational power limits on the Jetson Nano, we use a lightweight version of YOLOv4, YOLOv4-Tiny, which we trained on two custom data-sets and then optimise for deployment on the Jetson Nano [47]. The first data-set contained 10000 random images with labelled bounding boxes from the *Person* class of the Open Images dataset (OID) [48]. The second contained 500 photos from the *Cheetah* class from OID and 10000 hand-labelled images of cheetahs - extracted from videos taken at the Ann Van Dyk Cheetah Centre (De Wildt, South Africa) and Cheetah Outreach (Somerset West, South Africa) for [41].

The *Person* network was trained for 12800 iterations on 80% of the data-set, with 20% being used for validation. The *Cheetah* network was pre-trained for 1000 iterations on the 500 OID cheetah images and then trained for 11000 iterations on 80% of the hand-labelled cheetah data, with the other 20% being used for validation.

YOLOv4-Tiny's output is a set of bounding box coordinates and confidence scores. If there are multiple detected objects on initialisation, we pick the detected object with the highest confidence and then in subsequent frames we select the detected object whose center in that frame is closest to the center of our subject in the previous frame. We extract the x-position x_{CBB} (pixels) of the center of the tracked subject's bounding box in the webcam frame at a rate of 10 Hz.

2) Motor Control: Since the Maxon 50/5 servo controller only allows for closed-loop speed and current control, we needed to be able to control the motor's position as a function of the current. For this reason, we modelled the motor in state space (with an augmented integrator state) as follows:

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\\ e\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ e\end{bmatrix} + \begin{bmatrix}0\\ \frac{NK_t}{J}\\ 0\end{bmatrix}\begin{bmatrix}i\end{bmatrix} \tag{1}$$

$$y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ e\end{bmatrix}, \tag{2}$$

where θ and $\dot{\theta}$ are the motor shaft position and velocity, e is an added state for the integral of the error, N is the gear ratio, K_t is the motor torque constant, J is the combined motor and load inertia, and i is the motor armature current.

Using MATLAB Control System Toolbox the gains were chosen to be [6.0918 0.9208 14.6307]. The position controller executes at 50 Hz with a target position from the EKF and sends the desired motor current to the Maxon 50/5 controller (set in current control mode). Sensors and motor control including EKF are handled on a Teensy 4.0. We use FSF with an integral state (gains as reported above). This formulation avoids the derivative-noise sensitivity and re-tuning burden of a classical PID under irregular (10 Hz) measurement updates and timestamp jitter, while preserving fast re-centering at 50 Hz actuation.

3) Extended Kalman Filter: Due to hardware limitations, the object detection neural network cannot run at the same frequency as the FSF controller. To avoid this, we use an EKF with a sample time of 0.02 seconds with a constant

acceleration model and define the state vector as follows:

$$\hat{x} = \begin{bmatrix} \ddot{\theta} & \ddot{\varphi} & \dot{\theta} & \dot{\varphi} & \theta & \varphi \end{bmatrix}, \tag{3}$$

where θ and φ are the motor and subject's angular positions in the world frame (see Fig. 4 below). φ is the target position that is sent to the FSF controller as depicted in Fig 4. We measure θ directly using the encoders, and calculate φ from x_{CBB} using a non-linear measurement function that converts from pixels to radians through imaging geometry as shown below:

$$h(\hat{x}) = f \tan(\varphi - \theta), \tag{4}$$

where f is the webcams focal length in pixels.

The measurement co-variance of YOLOv4-Tiny was estimated to be $\tilde{}5^2$ $pixels^2$ by comparing YOLOv4-Tiny bounding box centers to hand-labelled bounding box centers. Due to the high resolution of the encoders, we set the encoder measurement co-variance to a low value of $0.01^2\,\mathrm{rad}^2$. For process noise we scaled the maximum expected jerk between samples for each set of states (acceleration, velocity and position), with the acceleration states having the least noise and position states having the most noise.

The EKF runs at 50 Hz and incorporates the high-rate motor encoder at every step; YOLOv4-Tiny detections arrive at 10 Hz and are fused when available via the non-linear pixel-to-angle measurement in (4). To remain stable through multi-frame detector outages, if no new detection arrives for more than 10 samples (≈200 ms) we inject a zero-mean pseudo-observation on the subject azimuth, $z_\varphi \sim \mathcal{N}\big(0, (\pi/4)^2\big)$. This prevents unbounded covariance growth during gaps while avoiding bias when the subject is genuinely off-center. The EKF output $\hat{\varphi}$ serves as the position setpoint to the FSF loop in Sec. III–B.2, forming a simple $10\,\mathrm{Hz} \rightarrow 50\,\mathrm{Hz}$ temporal bridge that keeps the controller responsive despite sparse/noisy detections.

The practical lower limit on detector rate is not a fixed number, but depends on the subject's angular speed and the closed-loop recentering bandwidth of the rig. In our setup, 10 Hz detections are bridged to 50 Hz control, so each detector update spans about five control steps and the controller remains responsive. As the detector rate decreases, the predicted target motion between updates grows, and performance degrades once that inter-update motion becomes large relative to the camera field of view or the controller settling time. In that regime, lower detector rates can still be used for slower subjects, but faster motion would require either a stronger motion model, more predictive control, or a faster detector.

The main failure modes are prolonged detector dropouts, short-term occlusions, identity swaps when multiple candidate subjects are present, and downstream 2D keypoint failures. During short detector outages, the EKF prediction and the pseudo-observation keep the estimate bounded, so the controller degrades conservatively rather than diverging. When reliable detections resume, the normal EKF update automatically re-locks the target estimate without requiring a manual reset. Very long outages or incorrect re-identification can still lead to temporary loss of centering, so reacquisition quality ultimately depends on the detector and the nearest-neighbor association heuristic.

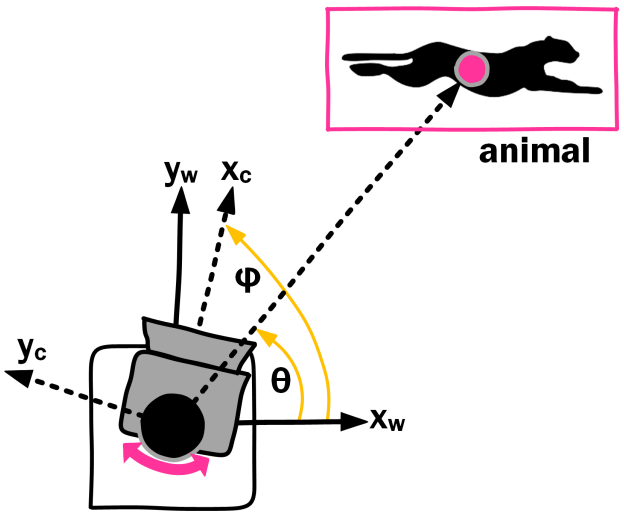

Fig. 4. Figure showing the coordinate axes and states for the motor control and EKF.

C. Pose Estimation

For 3D reconstruction, we use a trajectory optimization method for 3D pose estimation called FTE, developed in [41]. The FTE method fits a skeletal model for the subject species to a set of $2D$ pose measurements over a given trajectory so as to minimise both the model error and measurement error.

Naturally, the number of keypoints and degrees of freedom vary from subject to subject, but the generalised coordinates follow a common structure:

- x, y, z: The position of the first or base keypoint in the world frame.
- $\phi_1 ... \phi_L$: The roll angles of each subsequent keypoint, where L is the total number of joints
- $\theta_1 ... \theta_L$: The pitch angles of each subsequent keypoint
- $\psi_1 ... \psi_L$: The yaw angles of each subsequent keypoint

These coordinates represent the pose parameters of the kinematic model, and are optimized over a given trajectory to obtain a set of optimized states for each frame.

In the construction of our optimization problem, a cost function including both the measurement and model errors must first be formulated. As in our original work, we define the measurement error e_{meas} as:

$$e_{meas} = \sum_{i=1}^{n} \sum_{j=1}^{c} \sum_{k=1}^{m} \sum_{l=1}^{2} C\left(\frac{\mathbf{v}_{i,j,k,l}}{\sigma_{meas}} \right), \tag{5}$$

where $\mathbf{n}$ is the frame number, $\mathbf{c}$ is the index of the camera, $\mathbf{m}$ is the joint of the skeletal model, and $\mathbf{2}$ refers to the two dimensions $\mathbf{x}$ and $\mathbf{y}$. We divide by the standard deviation of the $2D$ measurements σ_{meas} to normalize the measurement

cost. Here, $C(\cdot)$ refers to an outlier-rejecting re-descending cost function as described in [41].

The model error e_{model} is chosen as the deviation from a constant acceleration model. The acceleration error $\mathbf{w}$ is squared and divided by the measured variance of the corresponding pose parameter $\mathbf{p}$. This provides us with the model error:

$$e_{model} = \sum_{i=1}^{n} \sum_{j=1}^{p} \frac{\mathbf{w}_{i,j}^2}{\sigma_{model\,j}^2} = \sum_{i=1}^{n} \sum_{j=1}^{p} \left(\frac{\mathbf{w}_{i,j}}{\sigma_{model\,j}} \right)^2. \quad (6)$$

We may then simply minimize the sum of the measurement and model errors over the whole trajectory. Our total cost function becomes:

$$\min_{\mathbf{x},\,\dot{\mathbf{x}},\,\ddot{\mathbf{x}}} \quad e_{meas} + e_{model}. \quad (7)$$

No additional scalar weighting was introduced between e_{meas} and e_{model} because both terms are already normalized by their corresponding standard deviations in (5) and (6), making the summed objective dimensionless and consistent with a standard Gaussian error interpretation. While this is the same process developed in [41], here we demonstrate its effectiveness in the 3D reconstruction of not only cheetahs, but humans and other species as well.

To compensate for the rotation of the cameras about the rig axis, we obtain FTE keypoint outputs relative to the camera frame and rotate these 3D coordinates about the vertical axis ($\hat{\mathbf{k}}$) to obtain world frame keypoint outputs. The magnitude of the rotation is given by the recorded rotary encoder value for the given frame.

IV. Results

Across our experiments, rotating cameras captured more frames with the subject visible than static cameras: **human +52% (6593 vs. 4333)**; **cheetahs +135% (2419 vs. 1031)**. Histograms show a strong center-bias with rotation (majority of frames within the central 20% of pixels; see Fig. 5) versus an even spread for static cameras, indicating successful centering and reduced off-axis time. These two effects—*more frames in view* and *more frames near center*—are expected to improve downstream 2D and 3D accuracy for a wide range of pose-estimation backends.

A. Object Detection

The performance of object detection neural networks is judged according to their Mean Average Precision (mAP). Average precision is a metric that combines precision and recall, with precision being the ratio of true positive area to total positive area, and recall being the ratio of true positive detections to total objects. The mAP is usually stated for a given threshold or Intersection of Union (IoU), which is the area of overlap between the inferred bounding box and the ground truth bounding box divided by the area of the union. For the *Person* network, we achieved a mAP of 36.01% with an IoU threshold of 0.5. For the *Cheetah* network, we achieved an mAP of 34.46% with the same IoU threshold.

Inference is done on the incoming 1920×1080 (1080p) video-stream from the webcam. Average inference time is 0.043 seconds on the Jetson Nano. This means that a theoretical maximum of 23.3 frames per second (fps) could be achieved for the object detection stage. This is limited to 10 fps to conserve resources so that additional sensors can be connected to the Jetson Nano at a later stage.

During testing we recorded the total number of frames processed and the number of frames in which nothing was detected (while the subject was in frame). We found that with a threshold of 0.5 the subject was missed by YOLOv4-Tiny in ˜2% of the frames where the subject was within a range of 15m (for both the *Person* and *Cheetah* network). This metric is subjective and depends on the completeness of the training data-set, occlusions, lighting and a variety of other factors. We save the frames with no detections so that these can later be manually labelled and used to augment the training dataset.

B. Subject Tracking

We tested the tracking system on a human subject running back and forth perpendicular to the tracking system at distances of 5m, 9m and 12m away. We also tested the tracking system on three free-running cheetahs (Amelia, Elliot and Kiara) at the Ann van Dyk Cheetah Centre. The cheetahs ran perpendicular to the system at approximate distances of 8m and 12m. To validate the tracker's performance in 3D reconstruction, two GoPro Hero Session 5 cameras were attached to the static base of the tracker and two more were attached to the rotating sensor pedestal. The image data collected by the GoPros during these experiments is summarized in the table below and the histogram in Fig. 5.

TABLE I
Total Number of Frames with Subject in View

Subject	R Cam 1	R Cam 2	S Cam 1	S Cam 2
Human	3306	3287	2188	2145
Cheetah	1206	1213	519	512

Note that R Cam and S Cam refer to the cameras attached to the rotating (R) sensor pedestal and the ones attached to the static (S) base respectively. In the human experiment, a total of 6593 frames were captured with the subject in frame of the rotating cameras, compared with 4333 between the two static cameras. This shows a $52.16\,\%$ increase in the amount of data collected due to the tracking system. Over the three cheetah experiments, a total of 2419 frames were captured with the subject in the rotating cameras' fields of view and only 1031 with the subject in the static cameras' fields of view, showing a $134.63\,\%$ increase in data collected.

The histogram in Fig. 5 shows the distribution of where the subject was in the camera frame for each frame captured. With the static cameras the subject is evenly located throughout the image frame, while for the rotating cameras the subject's center is within the center 20% of pixels in the majority of the captured frames. The cheetahs' positions in the images are skewed slightly to one side, indicating mild

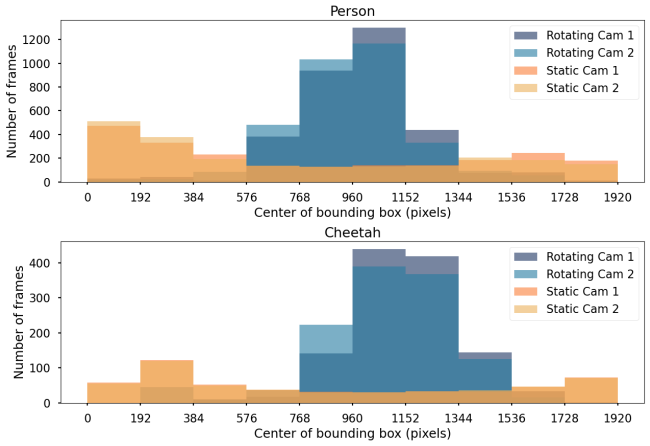

Fig. 5. Histograms showing distribution of subject location in image frames for the human and cheetah experiments (10 bins)

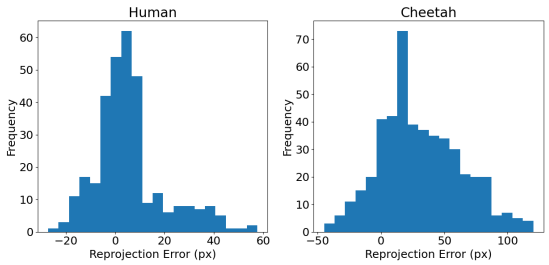

Fig. 6. Histograms showing reprojection errors in px for the human and cheetah subjects (20 bins)

tracking lag relative to the target position. This suggests that controller retuning or more anticipatory control strategies may further improve centering for faster subjects, but we do not claim that gain tuning alone is sufficient from the present experiments.

C. Pose Estimation

1) Quantitative Results: The final output of our method for 3D motion capture is a set of states describing the absolute positions of each keypoint on the subject species in 3D space. From these, we may easily obtain the coordinates x, y, and z of each keypoint in the world frame.

For the purposes of quantitative evaluation, we make use of a reprojection error. This was done by reversing the rotational compensation to obtain 3D keypoint positions relative to the rotating cameras, and using the previously obtained camera extrinsics and intrinsics to reproject each 3D keypoint to the 2D camera plane. Here, hand-labelled 2D ground truth data for the pose of each subject was compared with the reprojected FTE pose estimates. For the sample human trajectory, $n = 560$ hand-labelled points were considered. For the cheetah trajectory, $n = 800$ points were used.

Shown below are the normalized root mean square error (NRMSE) and percentage of correct keypoints (PCK) for the human and cheetah subjects:

TABLE II
NORMALISED RMSE AND PCK FOR EACH SUBJECT

Subject	NRMSE	PCK
Human	0.28	0.88
Cheetah	0.48	0.76

The NRMSE is obtained by dividing the ordinary RMSE by $\sqrt{h * w}$, where h and w are the height and width of the subject bounding box. The bounding box is also used to calculate the PCK. A keypoint is considered "correct" when the error is less than $\alpha * \sqrt{h * w}$, where α is a fraction chosen according to the average subject size and shape in the image plane. For the human trajectory we used $\alpha = 0.1$ while for the cheetah trajectory, we used $\alpha = 0.2$.

Shown below are histograms of the reprojection errors in pixels (px) for each of the subjects:

2) Qualitative Results: To provide a qualitative comparison, Fig. 7 shows representative original frames together with the corresponding world-frame FTE reconstructions. As seen in Fig. 7 and the supplementary video, despite the cameras rotating we are able to adequately produce plausible 3D pose trajectories for both test subjects.

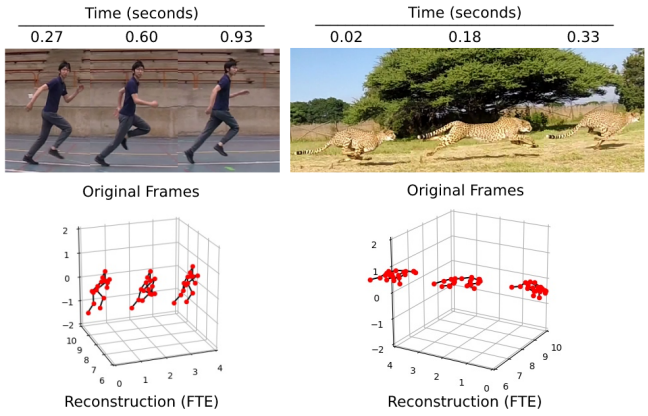

Fig. 7. Representative original frames (top) and corresponding FTE reconstructions in the world frame (bottom) for the human and cheetah sequences. The bottom plots are shown in a common world frame for qualitative interpretation and are not image-plane overlays.

3) Why centering helps any backend (analytic intuition): Let σ_{2D} denote the 2D keypoint error (pixels), s the subject pixel scale (e.g., $s = \sqrt{w\,h}$ from a bounding box of width w and height h), and f the focal length in pixels. Under a standard perspective model, the depth variance scales as $\mathrm{Var}(Z) \propto \dfrac{\sigma_{2D}\, Z^2}{f\, s}$; likewise, multi-view triangulation error decreases with larger s and stable baselines. Autonomous centering increases s and reduces partial occlusions, improving 2D PCK and 3D reprojection error *independent* of the specific 3D reconstruction backend (e.g., FTE). This provides a mechanistic link from our yield/centering results to downstream accuracy without further experiments.

V. Conclusion

At the current accuracy level, the system is best viewed as a field-deployable data-acquisition tool rather than a replacement for laboratory-grade motion-capture infrastructure. It is well suited to applications such as collecting longer usable trajectories, increasing the yield of reconstructable field footage, building training datasets, and supporting coarse-to-moderate kinematic analyses of locomotion and behavior. In contrast, applications that require very high precision joint kinematics or clinically definitive measurements would still benefit from tighter calibration, higher-resolution sensing, or more constrained capture conditions.

We showed that a simple, autonomous rotating-camera system substantially increases usable data yield for outdoor 3D animal motion capture and integrates seamlessly with existing reconstruction pipelines. A 10 Hz detector bridged to a 50 Hz FSF controller via an EKF keeps uncooperative subjects centered, requiring no human operator and only commodity hardware. On field videos, rotating cameras captured *more frames* and *more centered frames* than static cameras for both a running human and free-running cheetahs, and our rotation-aware FTE recovered plausible 3D trajectories. Looking ahead, auto-zoom and multi-modal sensing can compound these gains, but even without them, active centering is a practical lever to scale wildlife motion capture in the wild. More broadly, the platform is intended to complement existing ecological monitoring and animal-motion analysis pipelines by improving the fraction of field footage that is usable for downstream reconstruction.

Acknowledgment

We thank Ines and Mikayla at the Ann van Dyk Cheetah Centre for their help, and Harry at Cheetah Outreach for his continued support and assistance.

References

[1] E. Lurie-Luke, "Product and technology innovation: What can biomimicry inspire?," Biotechnology Advances, vol. 32, no. 8, pp. 1494–1505, 2014.

[2] A. Patel, "Understanding the motions of the cheetah tail using robotics," Ph. D. dissertation, Dept. of Elec. Eng, Univ. of Cape Town, Cape Town, 2015.

[3] A. Mathis, S. Schneider, J. Lauer, and M. W. Mathis, "A Primer on motion capture with Deep Learning: Principles, pitfalls, and Perspectives," Neuron, vol. 108, no. 1, pp. 44–65, 2020.

[4] L. Mündermann, S. Corazza, and T. P. Andriacchi, "The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications," Journal of NeuroEngineering and Rehabilitation, vol. 3, no. 1, 2006.

[5] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt, "Performance capture of interacting characters with handheld kinects," in Proc. Eur. Conf. Comput. Vis., 2012, pp. 828–841.

[6] L. Xu et al., "FlyCap: Markerless Motion Capture Using Multiple Autonomous Flying Cameras," in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 8, pp. 2284-2297, 1 Aug. 2018, doi: 10.1109/TVCG.2017.2728660.

[7] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel, "Markerless motion capture with unsynchronized moving cameras," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 224–231.

[8] "Vicon systems", [online] Available: http://www.vicon.com.

[9] R. Raskar et al., "Prakash: Lighting aware motion capture using photosensing markers and multiplexed illuminators", ACM Trans. Graph., vol. 26, no. 3, 2007

[10] L. A. Schwarz, D. Mateus and N. Navab, "Multiple-activity human body tracking in unconstrained environments" in Articulated Motion and Deformable Objects, Berlin, Germany:Springer, pp. 192-202, 2010.

[11] D. Vlasic et al., "Practical motion capture in everyday surroundings", ACM Trans. Graph., vol. 26, no. 3, 2007.

[12] P. Zhang, K. Siu, J. Zhang, C. K. Liu and J. Chai, "Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture", ACM Trans. Graph., vol. 33, no. 6, 2014.

[13] K. Foss, R. C. da Costa, and S. Moore, "Three-dimensional kinematic gait analysis of Doberman Pinschers with and without cervical spondylomyelopathy," Journal of Veterinary Internal Medicine, vol. 27, no. 1, pp. 112–119, 2012.

[14] K. M. Chu, S. H. Seto, I. N. Beloozerova, and V. Marlinski, "Strategies for obstacle avoidance during walking in the cat," Journal of Neurophysiology, vol. 118, no. 2, pp. 817–831, 2017.

[15] 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

[16] 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

[17] 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.

[18] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: 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.

[19] M. Burenius, J. Sullivan, and S. Carlsson, "3d pictorial structures for multiple view articulated pose estimation," in Proceedings ofthe IEEE Conference on Computer Vision and Pattern Recognition, pp. 3618–3625, 2013.

[20] 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.

[21] 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.

[22] A. Arnab, C. Doersch, and A. Zisserman, "Exploiting temporal context for 3d human pose estimation in the wild," in Proceedings ofthe IEEE Conference on Computer Vision and Pattern Recognition, pp. 3395–3404, 2019.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] B. E. Jackson, D. J. Evangelista, D. D. Ray, and T. L. Hedrick, "3D for the people: Multi-camera motion capture in the field with consumer-grade cameras and open source software," Biology Open, vol. 5, no. 9, pp. 1334–1342, 2016.

[28] 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.

[29] T. D. Pereira, D. E. Aldarondo, L. Willmore, M. Kislin, S. S.-H. Wang, M. Murthy, and J. W. Shaevitz, "Fast animal pose estimation using Deep Neural Networks," Nature Methods, vol. 16, no. 1, pp. 117–125, 2018.

[30] P. Karashchuk, K. L. Rupp, E. S. Dickinson, S. Walling-Bell, E. Sanders, E. Azim, B. W. Brunton, and J. C. Tuthill, "Anipose: A toolkit

for robust markerless 3D pose estimation," Cell Reports, vol. 36, no. 13, 2021.
[31] T. L. Hedrick, "Software techniques for two- and three-dimensional kinematic measurements of biological and Biomimetic Systems," Bioinspiration & Biomimetics, vol. 3, no. 3, p. 034001, 2008.
[32] W. I. Sellers and E. Hirasaki, "Markerless 3d motion capture for animal locomotion studies," Biology open, vol. 3, no. 7, pp. 656–668, 2014
[33] 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
[34] 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
[35] 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
[36] T. Nakamura, J. Matsumoto, H. Nishimaru, R. V. Bretas, Y. Takamura, E. Hori, T. Ono, and H. Nishijo, "A markerless 3D computerized motion capture system incorporating a skeleton model for monkeys," PLOS ONE, vol. 11, no. 11, 2016.
[37] P. C. Bala, B. R. Eisenreich, S. B. 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, 2020.
[38] L. Clark, "Markerless 3D Motion Capture of Cheetahs in the Wild," MSc. dissertation, Dept. of Elec. Eng, Univ. of Cape Town, Cape Town, 2021.
[39] S. Raman, R. Maskeliūnas, and R. Damaševičius, "Markerless dog pose recognition in the wild using resnet deep learning model," Computers, vol. 11, no. 1, p. 2, 2021.
[40] I. Zelman, M. Galun, A. Akselrod-Ballin, Y. Yekutieli, B. Hochner, and T. Flash, "Nearly automatic motion capture system for tracking octopus arm movements in 3d Space," Journal of Neuroscience Methods, vol. 182, no. 1, pp. 97–109, 2009.
[41] D. Joska, L. Clark, N. Muramatsu, R. Jericevich, F. Nicolls, A. Mathis, M. W. Mathis, and A. Patel, "AcinoSet: A 3D pose estimation dataset and baseline models for cheetahs in the wild," 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021.
[42] R. Labuguen, J. Matsumoto, S. B. Negrete, H. Nishimaru, H. Nishijo, M. Takada, Y. Go, K.-ichi Inoue, and T. Shibata, "MacaquePose: A novel 'in the wild' macaque monkey pose dataset for Markerless Motion Capture," Frontiers in Behavioral Neuroscience, vol. 14, 2021.
[43] Y. Wang, Y. Liu, X. Tong, Q. Dai and P. Tan, "Outdoor markerless motion capture with sparse handheld video cameras," in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 5, pp. 1856-1866, 1 May 2018, doi: 10.1109/TVCG.2017.2693151.
[44] C. Wu, C. Stoll, L. Valgaerts and C. Theobalt, "On-set performance capture of multiple actors with a stereo camera", ACM Trans. Graph., vol. 32, no. 6, pp. 161, 2013.
[45] M. V. Srinivasan, H. D. Vo, and I. Schiffner, "3D reconstruction of bird flight trajectories using a single video camera," bioRxiv, 01-Jan-2022. [Online]. Available: https://doi.org/10.1101/340232. [Accessed: 24-Feb-2022].
[46] E. de Margerie, M. Simonneau, J.-P. Caudal, C. Houdelier, and S. Lumineau, "3D tracking of animals in the field, using rotational stereo videography," Journal of Experimental Biology, 2015.
[47] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal Speed and accuracy of object detection," arXiv.org, 23-Apr-2020. [Online]. Available: https://arxiv.org/abs/2004.10934. [Accessed: 01-Mar-2022].
[48] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The Open Images Dataset V4," International Journal of Computer Vision, vol. 128, no. 7, pp. 1956–1981, 2020.