Context-Aware Outlier Rejection for Robust Multi-View 3D Tracking of Similar Small Birds in An Outdoor Aviary

Keon Moradi
University of Oklahoma
keon.moradi@ou.edu

Ethan Haque
University of Oklahoma
ethan.a.haque-1@ou.edu

Jasmeen Kaur
University of Oklahoma
jkaur@ou.edu

Alexandra Bentz
University of Oklahoma
abentz@ou.edu

Eli Bridge
University of Oklahoma
ebridge@ou.edu

Golnaz Habibi
University of Oklahoma
golnaz@ou.edu

Abstract

This paper presents a novel approach for robust 3D tracking of multiple birds in an outdoor aviary using a multi-camera system. Our method addresses the significant challenges posed by the visual similarity of birds and the complexities of their rapid movements in three-dimensional space. We leverage environmental context, called landmarks, to enhance feature matching between cameras, leading to more accurate 3D reconstructions and tracking of birds positions. Removing outliers in multi-view settings is especially challenging because the objects of interest are visually similar. In our approach, outliers are rejected based on their nearest landmark. This leads to constructing precise 3D models of individual birds; additionally, it applies these models to track multiple birds simultaneously. By utilizing environmental context, our approach significantly improves the differentiation between visually similar birds, a key obstacle in existing tracking systems. Experimental results demonstrate the effectiveness of our method, showing a 20% *elimination of outliers in the 3D reconstruction process, with an accuracy of* 97% *accuracy in matching. This remarkable accuracy in 3D modeling translates to robust and reliable tracking of multiple birds, even in challenging outdoor conditions. Our work not only advances the field of computer vision but also provides a valuable tool for studying bird behavior and movement patterns in natural settings. We also provide a large annotated dataset of 80 birds residing in four enclosures for 20 hours of footage which provides a rich testbed for researchers in computer vision, ornithologists and ecologists.*

1. Introduction

Characterizing complex behaviors among animals living in naturalistic settings is crucial in biological and ecological science. This requires fine-scale behavioral tracking that is not readily available. Despite the advances of computer vision in the field of advanced mobility and robotics, there are not extensive studies that apply advanced 3D computer vision to tracking the animals consistently and reliably. Most current works are not generalizable to new environments. Moreover, much of them are limited to 2D detection and tracking, which leads to missing the tracks of animals with rapid movement such as flying birds and bats. Multi-view technology has been used recently in the application of computer vision to provide accurate 3D construction for detection and tracking of objects in a 3D world. Multi-view tracking has also advanced robotics and autonomous driving by providing affordable sensing techniques compared to LiDAR technology. However, using this 3D computer vision technology in tracking highly dynamic animals such as birds and bats is not trivial. Developing methods that can accurately capture and analyze complex flight behaviors and social interactions in settings that closely mimic natural environments remains challenging. Behavioral biologists have historically used continuous observations of one focal animal for scan samplings of groups [3]. These approaches are labor-intensive and only capture a small subset of important social behaviors for a few individuals at a time. To record behavioral data at a finer scale, simultaneously across all individuals in a population, more advanced behavioral tracking technology is needed. Recent advances in computer vision and deep learning have made this a reality; however, to-date these techniques have largely been restricted to laboratory model species [10] and mammals [2, 8, 13], while work in avian species is still limited [6]. Birds are conspicuous, largely diurnal, and our knowledge

of avian natural history and ecology is more extensive than that of any other vertebrates [11] Thus, birds could offer unique insights into complex social behaviors if we could appropriately track them in naturalistic settings.

However, tracking the birds in an outdoor environment is challenging due to their similarity, small size, and different background. In this paper we focus on detecting and tracking identical birds, and we investigate and integrate combinations of techniques that optimize 3D-reconstruction results within the containment of an aviary environment.

1.1. Main Contribution

The main contributions of this paper are listed as follows:

- Propose a robust multi-view multi object tracking of visually similar birds in an aviary with natural background
- Improve feature matching using landmarks - outlier rejection
- Propose a novel outlier rejection algorithm which is based on the environmental context, called landmarks, to achieve an accurate 3D reconstruction for robust tracking.
- Results show our approach eliminates limitations in the previous work in terms of re-IDentification of the birds, or mismatching birds, across multiple images, leading to more consistent tracking, less ID switching, and less numbers of missing tracks, all of which improves the overall quality of tracking.
- Provide a large, annotated dataset for further study of animal behavior and spatio-temporal tracking.

2. Related Work

Tracking animals, especially birds, produces several challenges such as occlusions, being very similar to surroundings or each other, as well as having three-dimensional movement challenges. However, in recent years, tracking animals in the computer vision world has become a rising field in many studies. They have attempted to address these previously mentioned issues through multi-camera systems and computer vision techniques, which are very foundational to our approach and our solution to these issues pertaining to tracking birds.

[17] presented a system for teaching and analyzing songbirds within a multi-view 3D aviary. Their work primarily observed occlusion and a variety of appearances of these songbirds in the 3D space, in which they used stereo matching and multi-view tracking. They also utilized a challenging dataset called WILD, which made the creation of their techniques in combining detection softwares like Mask-RCNN, as well as a Background Subtraction mask, more novel. This later also becomes crucial in our techniques of segmentation. However, while they relied on stereo matching for 3D reconstruction, our approach relies more on the context of the birds, specifically its context-aware outlier rejection which improves the accuracy of feature matching and 3D reconstruction.

Furthermore, [16] worked with bats and wanted to test which approach for tracking them better was more resourceful, either Reconstruction-then-Tracking(RT) or Tracking-then-Reconstruction(TR) in complex environments. Their findings show that the RT methods actually perform better in datasets that have a lot occlusions which is another reason why we select the RT approach as our birds sometimes reside in highly occluded areas, and we extend this by adopting a landmark-based matching system to be able to hand these occlusions as well as similarities.

These are some of the works that have been crucial in us exploring our techniques, and several of these animal tracking studies have primarily been in controlled lab environments such as [1] and [7]. Our work is in a more dynamic outdoor setting, where lighting situations are exactly of those in the real world, and we can tackle head on challenges of tracking small animals, such as birds. By incorporating our landmark-based outlier rejection, we challenge the difficulties of matching identical birds that look indistinguishable from each other, while also reducing ID switches and obtaining more consistent 3D tracking.

3. Dataset

We have provided an aviary dataset which includes 20 hours of footage of 80 birds in an outdoor aviary. We utilize five GoPro cameras (1920x1080 pixels, 30 FPS) aligned at various positions throughout an outdoor aviary enclosure (27.2 m^3). The aviary is enclosed in hardware cloth and subjected to natural light cycles. Video recordings were synced using GoPro software to ensure proper alignment. The MP4 files are subsequently chopped into JPG frames based on the 30 FPS frame-rate of the GoPro cameras, and 120 frames from each camera view are manually annotated with boxed regions that reflect bird detections, to be used for training the YoloV5 model to obtain bird detections on the rest of the video frames.

The subjects were 20 adult house sparrows (*Passer domesticus*) in 4 enclosures, totaling 80 birds, captured locally and placed in mixed-sex flocks. Each bird was given a unique combination of colored plastic leg bands for visual identification. Food and water were provided throughout the aviary in bowls, and wooden nest boxes were placed along the walls. Five corresponding camera views are studied in this paper and are shown in Fig 1.

(a) Camera view 1 (b) Camera view 2 (c) Camera view 3 (d) Camera view 4 (e) Camera view 5

Figure 1. Five camera views in one of the enclosures in the aviary, which is studied in this project.

4. Multi-View Multi Bird Tracking Overview

Fig 2 depicts the pipeline of our workflow for 3D tracking of birds in an aviary using multiple cameras. As illustrated, the proposed algorithm has eight main steps: **(1) Object detection:** The workflow of this project begins operation on initially synced videos passed through our video processing pipeline, which initially chops the videos into image frames. Then, 120 frames per camera view are manually annotated with bird detections in their frame to use as training data for the YoloV5 model. [14] After training, this model is then used to obtain bird detections throughout the remaining image frames corresponding to the videos that were initially passed in, and .csv files storing the bounded box coordinates of bird detections and other relevant data are saved; **(2) Masking:** our image masking segmentation is applied to image frames to obtain a binary mask depicting bird detection and background features from that frame; **(3) Keypoint Extraction:** a number of keypoints and corresponding descriptors are extracted from "on" pixel regions of the mask using the SIFT keypoint extraction algorithm [12]; **(4) Feature Matching:** the extracted keypoints are used to match birds and bird features between camera views to help globally track similar birds across all camera views using the Brute-Force matching algorithm; **(5) Outlier Rejection:** we integrate our context-based outlier rejection method by constructing a Voronoi diagram using landmarks selected throughout our camera views as Voronoi coordinates constructing the Voronoi diagram, which is then layered on top of our image. We then use these 5 initialized Voronoi diagram objects (one per camera) to associate each bird to their nearest landmarks in each video frame. Then, of the initial features/keypoints matched earlier, only the ones that also have matching nearest landmarks (landmark closest to their corresponding bird detection) are validated and kept; **(6) Clustering:** by keeping the bounding box index (*i.e.,* bird) corresponding to each matching feature, the matched features are clustered; **(7) 3D reconstruction:** following the previous step, a 3D reconstruction representing the global view of our aviary is obtained using multi-view geometry and triangulation; **(8) Multi-object tracking:** each bird is tracked across five camera views by following its center in 3D coordinates. We utilize the tracking-by-detection technique, and we apply a Kalman filter to predict each bird's 3D position in the next frame.

5. Object Detection

Previous work [17] used Mask R-CNN [9] due to its ability to generate high-quality segmentation masks alongside object detection. However, considering the complexity and difficulties of our hardware and the large volume of data we need to process, Mask R-CNN proved to be less efficient. Instead, we use YOLOv5, which offers a more streamlined and faster approach to object detection while maintaining robust accuracy. The decision to switch to YOLOv5 was made due to its real-time processing capabilities and the need to handle high frame rates without sacrificing detection quality. YOLOv5's backbone architecture, coupled with its efficient detection head, allowed us to process each frame more rapidly, which was essential given our dataset's size and the dynamic nature of the aviary environment. Furthermore, as in Table 1, YOLOv5 yielded in more bird labels in our testing for detection models, which further made for a better base model to fine tune.

Initially, when running YOLOv5 on our dataset, it detected approximately 1448 birds in the first video which is only 16% of the birds. Recognizing the need for enhanced detection accuracy, especially given the challenging conditions of our aviary setup (e.g., occlusions, similar appearances of birds), we undertook a fine-tuning process. This involved augmenting the model with an additional 600 newly annotated frames specifically chosen to capture a variety of challenging scenarios.

The fine-tuning significantly improved the model's performance, with detections increasing to over 6000 in the same video—a nearly 7.5-fold improvement. This dramatic increase in detection counts proved to be fruitful, in addition to adjusting hyperparameters like learning rate and batch size. Continuous fine-tuning was also employed as new data became available, allowing the model to adapt to different lighting conditions, angles, and bird behaviors. This iterative process ensured that the model remained robust and could generalize well across different frames and scenarios.

As shown in Table 1, the trend of improvement across all of the different models and fine-tuning was done to receive the best possible results within the detection step before proceeding through our pipeline. We took the same video sample of Figure 1e and ran it through only 1 minute at 30FPS, or in total 1800 frames for each model, and we documented how many bird labels were predicted. We note

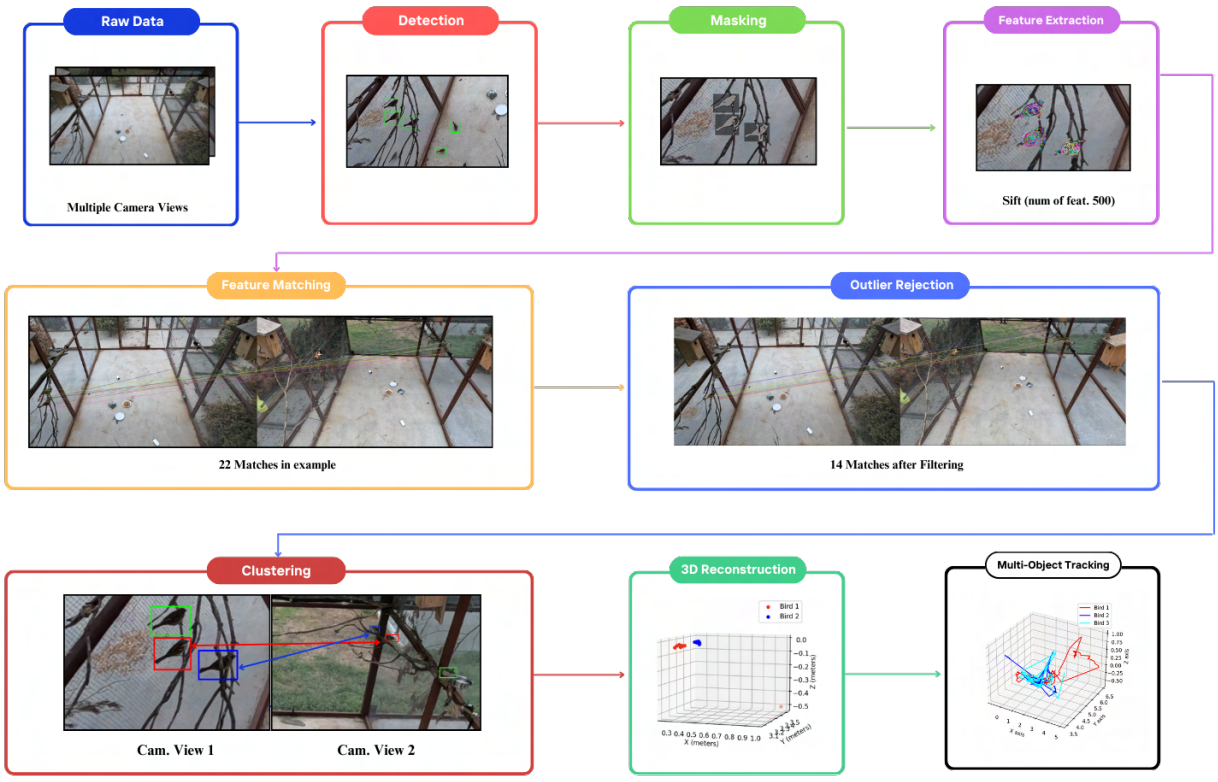

Figure 2. Proposed Workflow diagram: our multi-view 3D multi-bird tracking

that the Mask RCNN and YOLOv5 Base models detected many labels/classes other than just birds. Additionally, the numbers in Table 1 reflect the estimation of the number of birds detected. In fact, it is difficult to find an exact ground truth, due to the birds being blurry at times when in fast flight, as well as when there are detections on the side of the camera angle. The fine-tuning gave significant improvement to the detection step, especially the first 300 annotated frames that we trained on, and returned a closer estimate of detected bird labels after an additional 300 frames were annotated, totaling to 600 annotated frames upon which our YOLOv5 model was trained on.

Model/Method	Detected Bird Labels
Mask RCNN (Base Model)	823
YOLOv5 (Base Model)	1448
Fine-Tuned YOLOv5 (300 Annotated Frames)	5548
Further Fine-Tuned YOLOv5 (600 Annotated Frames)	6308
Estimate of Ground Truth (4-5 Birds, 1800 Frames)	7200 - 9000

Table 1. Detection results for different models in the bird tracking pipeline.

6. Feature Extraction and Matching

Image frames containing bird detections from the trained and tuned YOLOv5 model undergo our binary masking segmentation to obtain pixel regions that either directly or contextually indicate bird activity/relevant bird features. Our masking technique operates on image frames that contain bird detections by initially applying Canny edge detection on bounded boxes corresponding to bird detections [5].

Feature extraction is performed on masked image frames using the Scale-Invariant Feature Transform (SIFT) [12], which uses a filtering approach to only apply computationally expensive operations on locations in an image that pass an initial test. These heavier operations generate sets of features over images by first using a difference-of-Gaussian function $D(x, y, \rho)$ to identify suitable interest points in the image that are appear invariant to scale and orientation [12].

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (3)$$

At each potential location, a model is used to determine the measure of the corresponding points' invariability, and keypoints are chosen accordingly. Finally, local image gradients are used to assign orientations to each keypoint, and the combination of these orientations, scale, and location ensures invariability on all transformations and calculations performed on these features. Descriptors corresponding to each keypoint are subsequently obtained by transforming the local image gradients into representations that are suited to undergo significant levels of local shape distortion and change in illumination without corruption.

After obtaining a complete list of keypoints aligning with the bird detections in two camera views, we use Brute-Force feature matching combined with *k*-Nearest Neighbors algorithm to find correspondences between keypoints in images. After this step, we proceed to obtain calculations for the precision and recall of our matching results

7. Landmark based Outlier rejection

In the context of our environment, the challenge of matching extracted features between birds lies in the physical similarities of each bird with the rest. Even for humans, task of quickly identifying and matching birds in the aviary between camera views remains difficult due to the birds' dark color, similar shape, and frequently occlusive behavior. Though we are able to ensure that all extracted keypoints lie on birds by restricting ourselves to only scan over bounded-box coordinates, the similarity in all 20 birds' features limits the accuracy of keypoint matches generated by the Brute Force Matcher algorithm. To assist our efforts at improving match accuracy, we employ the use of Voronoi diagrams.

The Voronoi diagram of a set of points $P = \{p_1, ..., p_n\}$ in $\mathbb{R}^d$ creates a partition of $\mathbb{R}^d$ into n regions, where all points in a region share a common closest point in the set P according a distance metric $D(.,.)$. [15] Incorporating arbitrary distance metrics yields different variations of Voronoi diagrams. In other words, the region corresponding to a point $p \in P$ also contains all other points $q \in \mathbb{R}^d$ for which the following condition holds:

$$\forall p' \in P, p' \neq p, D(q, p) \leq D(q, p'). \quad (4)$$

Figure 3. (a) Keypoints in the first camera view; (b) Keypoints in the second amera view

By adapting the Voronoi diagram to be constructed from the local pixel coordinates of global landmarks across the aviary enclosure, we introduce a novel context-based outlier rejection algorithm for the previously obtained feature matches. We layer our initially constructed Voronoi diagram over each image frame, thus segmenting our image into regions uniquely identifiable by their local corresponding landmark. Figure 5 depicts this segmentation technique, where the blue dots correspond to the locations of landmarks selected in the camera view.

The Euclidean distance function measures the distance between two points $p, q \in \mathbf{R}^2$, $p = (p_1, p_2)$ and $q = (q_1, q_2)$ as such:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \quad (5)$$

By calculating the Euclidean distances between the pixel coordinates from each matched SIFT keypoints' corresponding bird and each local landmark in the camera view, we obtain the closest landmark identifier for each keypoint and its pair. If the two landmark identifiers do not match,

Figure 4. Masking Step. We consider masking algorithms that is shown in bottom row.

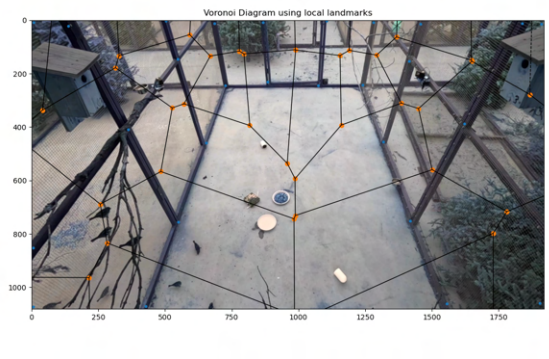

Figure 5. Locally constructed Voronoi diagram

we reject the match, using context-based inference to reject generally mismatched birds.

As Voronoi diagrams have not yet been used in this context of image segmentation, we were required to repurpose the existing Voronoi documentation in the SciPy Python library in order to ensure proper integration of the module for our tasks. We claim this as part of the novelty of our work, as we derived a process to subvert the creation of "infinite edges" within the current SciPy Voronoi documentation to ensure that all pixels in our image have a landmark corresponding to their Voronoi region.

8. 3D multi-object tracking of the birds

The process begins with the calibration of the cameras to determine the intrinsic and extrinsic parameters required for 3D Pose Estimation. We note that we needed to obtain the calibration's manually as this was not an automated process, and we used multiple methods such as OpenCV's Chessboard method [4] to match similar points and objects of known fixed positions across pairs of camera angles in order to find the calibration. For each camera, the intrinsic matrix K, distortion coefficients D, rotation vectors $rvecs$, and translation vectors $tvecs$ are used to compute the projection matrices. The projection matrix P for each camera is given by:

$$P = K[R|t] \tag{6}$$

where R is the rotation matrix obtained from $rvecs$, and t is the translation vector $tvecs$. The projection matrices allow us to map 2D image coordinates to 3D space.

Once the birds have been detected and matched across camera views, using the landmark-based feature matching described earlier, we proceed with the 3D reconstruction of their positions. For each matched pair of detections across two camera views, the 3D position of the bird is calculated using triangulation. This process involves finding the intersection point of the lines of sight from the two cameras. Given the matched 2D coordinates (x_1, y_1) and (x_2, y_2) from the two views, the 3D point $\mathbf{X}$ is obtained by solving the following equation:

$$\mathbf{X} = \text{triangulate}(\mathbf{x}_1, \mathbf{x}_2, P_1, P_2) \quad (7)$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the homogeneous coordinates of the matched points, and P_1 and P_2 are the projection matrices for the two cameras.

To ensure consistency and accuracy in tracking over time, we employ a Kalman filter with velocity tracking for each bird. The Kalman filter predicts the bird's next position in 3D space and updates this prediction based on new observations, thereby smoothing the trajectory and reducing noise. The state vector for the Kalman filter includes the position, velocity, and acceleration of the bird in 3D space.

In each frame, the Kalman filter predicts the 3D position of the bird. The predicted position is then compared to the new detections using Euclidean distance to find the best match. The Kalman filter is updated with the matched detection, refining the trajectory of the bird.

The integration of landmark-based matching, discussed in previous sections, plays a crucial role in the initial matching process, providing reliable correspondences between views that feed directly into the 3D triangulation and tracking pipeline.

8.1. Camera Calibration

We consider the triangulation and iterative reprojection process to automatically calibrate the cameras.

9. Experiments

In our experiments, we calculate the statistics pertaining to the quantity of keypoints extracted over our interval of frames, including the minimum and maximum number of keypoints extracted, and display them in Table 2. After initial keypoint extraction, we proceed with our context-aware outlier rejection step to eliminate incorrect matches. We calculate statistics pertaining to the validity of the results of this step in Table 3, and observe that the 0.97 ratio of correct final matches against all of the initial feature matches over the frame interval reflects positively on our outlier rejection methodology. Figure 6 shows the percentage of keypoints rejected in each frame across the length of our frame interval, and offers further insight into our problem domain. As previously mentioned, one main difficulty of tracking in this environment pertains to the extreme similarity in the physical features of the birds in the aviary, meaning that the descriptors obtained for matching birds are often extremely similar, and provide little meaningful information to distinguish individual birds from one another. Because of this, a large percentage of incorrect keypoint matches (approx. 80%) are expected to be removed from each frame, and this corresponds with the results in Figure 6.

Furthermore, we also set up a 3D reconstruction experiment for the purpose of testing our calculations of the 3D positions of the birds. We consider our reproduction error for this experiment, which measures how closely the reprojected 3D points back on to the 2D image aligned with the original 2D keypoints. We collected 30 frames of data over 3 different intervals and computed their metrics in Table 4. The average re-projection error reflects the averaged distance between the distance of the original 2D keypoints and their counterpart, which resulted in 19.8 average reproduction error. However, we note our low standard deviation, which shows that most errors were indeed close to the average error. Additionally, we present a percentage of keypoints below a threshold of 25 px error, which was at least 54.3% of the collected keypoint data, which can further illustrates our reconstruction quality.

Finally, we evaluated the performance of the tracking step in our pipeline, which used quantitative metrics seen in Table 5 that measure the consistency of our system. We conducted our tracking experiments over 3 different intervals, which had difficult situations for example multiple crossing over or when the birds go off screen for one camera and has high chance of an ID switch. Our first metric was the total count of ID switches across a 1 minute video, and see an average of 23.4 ID switches. Moreover, we counted how many birds were able to be tracked at certain time intervals.

Table 2. GoPro3 & GoPro5 Keypoint Statistics (frames 2200-2550)

Camera	GoPro3	GoPro5
# keypoints [min,max]	[2, 79]	[3, 81]
# keypoints (avg±std)	24±16	40±18

Table 3. Outlier Rejection Statistics for GoPro3 & GoPro5 pair

Avg feature match rejection %	79.03
Std. Dev feature match rejection %	20.45
Ratio correct final matches / all initial matches	0.20
Ratio correct final matches / all final matches	0.97

10. Conclusion

We have presented a multi-view pipeline to track visually similar birds in an outdoor aviary. We proposed a novel outlier rejection based on the environmental context using the Voronoi diagram. This Voronoi diagram is created based on the given landmarks in the aviary which can be defined by the user. The result shows our pipeline was able to achieve the matching accuracy of 97%. Using this feature matching, we were able to 3D construct the bird locations accurately. Our tracking algorithm shows a promising result, confirming a robust tracking of the birds. Testing our algorithm in different aviary datasets as well as combining it with animal behavior analysis are considered as a future work.

]

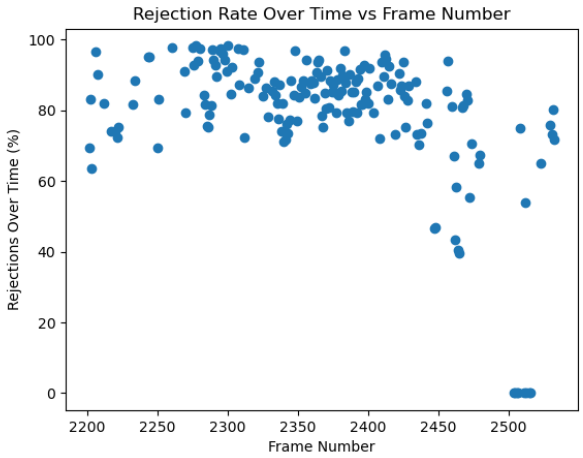

Figure 6. Outlier rejection statistics over frame interval

Table 4. Quantitative 3D Reconstruction Metrics and performance

Metric	Value
Reprojection Error (pixels)	
Total Keypoints	1,736
Average Reprojection Error	19.8
Standard Deviation of Error	5.3
Max Reprojection Error	36.2
Min Reprojection Error	9.7
% Keypoints Below 25px Error	54.3%

Note: The averaged metrics were across three intervals of 30 frames each.

Table 5. Quantitative Tracking Metrics for Bird Tracking System Performance

Metric	Value
Reprojection Error (pixels)	
Mean Reprojection Error (Camera 1)	14.32
Mean Reprojection Error (Camera 2)	5.03
Tracking Performance	
Total ID Switches (average)	23.4
Birds Tracked Over 10s (%)	77.1
Birds Tracked Over 30s (%)	56.7
Birds Tracked Over 60s (%)	26.7

Note: The metrics were averaged over three intervals of the same camera video.

References

[1] Marc Badger, Marc Schmidt, and Kostas Daniilidis. Automated 3d reconstruction of animal posture from video using keypoint detection and openpose. *arXiv preprint arXiv:2010.03832*, 2020. 2

[2] Praneet C Bala, Benjamin R Eisenreich, Seng Bum Michael Yoo, Benjamin Y Hayden, Hyun Soo Park, and Jan Zimmermann. Automated markerless pose estimation in freely moving macaques with openmonkeystudio. *Nature communications*, 11(1):4560, 2020. 1

[3] Melissa Bateson and Paul Martin. *Measuring behaviour: an introductory guide*. Cambridge university press, 2021. 1

[4] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008. 6

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):91–110, 1986. 4

[6] Cheng Fang, Tiemin Zhang, Haikun Zheng, Junduan Huang, and Kaixuan Cuan. Pose estimation and behavior classification of broiler chickens based on deep neural networks. *Computers and Electronics in Agriculture*, 180:105863, 2021. 1

[7] Adam Gosztolai, Semih Günel, Victor Lobato-Ríos, Marco Pietro Abrate, Daniel Morales, Helge Rhodin, Pascal Fua, and Pavan Ramdya. Liftpose3d, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals. *Nature methods*, 18(8):975–981, 2021. 2

[8] Adam Gosztolai, Semih Günel, Victor Lobato-Ríos, Marco Pietro Abrate, Daniel Morales, Helge Rhodin, Pascal Fua, and Pavan Ramdya. Liftpose3d, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals. *Nature methods*, 18(8):975–981, 2021. 1

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[10] Pierre Karashchuk, Katie L Rupp, Evyn S Dickinson, Sarah Walling-Bell, Elischa Sanders, Eiman Azim, Bingni W Brunton, and John C Tuthill. Anipose: A toolkit for robust markerless 3d pose estimation. *Cell reports*, 36(13), 2021. 1

[11] Masakazu Konishi, Stephen T Emlen, Robert E Ricklefs, and John C Wingfield. Contributions of bird studies to biology. *Science*, 246(4929):465–472, 1989. 2

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(6):91–110, 2004. 3, 4

[13] Markus Marks, Qiuhan Jin, Oliver Sturman, Lukas von Ziegler, Sepp Kollmorgen, Wolfger von der Behrens, Valerio Mante, Johannes Bohacek, and Mehmet Fatih Yanik. Deep-learning-based identification, tracking, pose estimation and behaviour classification of interacting primates and mice in complex environments. *Nature machine intelligence*, 4(4):331–340, 2022. 1

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 3

[15] Cyrus Shahabi, Mehdi Sharifzadeh, LING LIU, and M. TAMER ÖZSU. *Voronoi Diagrams*, pages 3438–3440. Springer US, Boston, MA, 2009. 5

[16] Zheng Wu, Nickolay Hristov, Sharon Swartz, Thomas Kunz, and Margrit Betke. Tracking-reconstruction or reconstruction-tracking? Technical Report BUCS-TR-2010-030, Boston University, Department of Computer Science, 2010. 2

[17] Shiting Xiao, Yufu Wang, Ammon Perkes, Bernd Pfrommer, Marc Schmidt, Kostas Daniilidis, and Marc Badger. Multi-view tracking, re-id, and social network analysis of a flock of visually similar birds in an outdoor aviary. *International Journal of Computer Vision*, 131(6):1532–1549, 2023. 2, 3