Baboons on the Move

Joshua Kang, Christie Wolters
University of California San Diego

Abstract

Baboons have proved to show that they are highly intelligent animals capable of a superb level of inter-troop communication and collaboration. We hope to enable researchers to study these amazing animals in the most active parts of their days where they have shown the highest level of cooperation, the morning (right after they wake up), and in the evening (right before they go to sleep). However, this raises the logistical question on having scientist observe these animals during the entire day for hours at a time. This study will introduce the mechanisms in which we have developed to be able to track individual baboons simultaneously through the use of continuous drone footage. We do not study the exact social interactions and behaviors of baboons in their respective troop and how they carry out tasks and cooperate with each other, but rather give the researchers a method in which they can carry out their discoveries in a more efficient way.

Keywords: Baboons, Kenya, Computer Vision, Particle Filter

1. Introduction

Collective and distributed decision-making has long been a topic of interest in animal research since it is a complex process in many nonhuman animal species. Long-lived social mammals that interact within societies have much in common with humans. Within these particular societies, individuals and their connections within their social network have a critical impact on group-level behavior. This is particularly true of nonhuman primates. In this paper, we examine tracking of baboon troop movements using a combination of human observers and computer vision techniques to aid in the study of the group-level behaviors that impact troop movement and collective decision-making.

The periods of decision-making (high volume shown in morning and evening) is relatively hard to study with current methods of manual field research, not only because of the sheer quantity of baboons involved but also because of the poor viewing angles of people taking notes on the ground. Any single researcher on the ground can see only a partial view of the big picture. With drone footage, not only can it be easier to skim through hours of irrelevant footage, as a computer can assist, but it can also be digitized and pieced together so that the group can be understood as completely as possible. The moment we are trying to define has multiple stages, including instances where smaller portions of the group or even individuals advocate for different directions, until they collectively decide on one final direction. Field researchers currently are uncertain of how they collectively make that decision, so by identifying when this is happening, making it easier to visualize, and giving insight into things we can't visualize (such as baboon direction and future trajectory prediction) we hope to make understanding their process much more feasible.

1.1. Current Work

The current work [1] on the project is using a technique called blob detection to localize random baboons at any given frame in the video. To the details that will not be included in this report, it leverages blobs that the baboons create, and gives us the centroids of the baboons that it can detect in that current frame. There are several issues with the current system that prevents it from being an effective tool for researchers :

- Discontinuous tracking
  The nature of the algorithm depends on the baboons to be 1) large enough for the blob detector to even see it, and 2) moving with enough motion from frame to frame. These are two problems that are quite significant for the success of the end-to-end tool. The former is a harder problem to solve, and can solved logistically (i.e using a higher resolution camera, getting closer to the ground). The latter poses a challenge which was the main motivation behind the next iteration that the lab has been experimenting on (to be discussed in 1.2). The current status quo is as simple as : 'If the baboon is not moving, we cannot track it'. This then introduces the second issue that of lack of individuality in tracking.

- Lack of individuality

  The seemingly small issue of discontinuity poses the question on how our platform can track a particular baboon throughout the entirety of the video. If we lose a baboon for even one frame, it is dangerous for us to answer the hypothesis as to which baboon it is when the blob detector discovers a baboon moving right where we last saw the baboon in consideration (could be 5-20 frames in discontinuity). Therefore, it raises the technical question on how we can leverage the high accuracy of the blob detector to track each individual baboon in the troop for the entirety of the video using filters (including baboons that appear in the video intermediately).

1.2. Filters

The Baboon Project takes drone footage captured at two main baboon settlements. It uses computer vision algorithms to identify where the baboons are in the footage and track individual baboons as they move around the settlement. The current algorithm that is used in the project is able to accomplish this, however, the time it takes to process the footage is at a rate of 20 minutes per 30 seconds of video. We will be attempting to reduce this processing time by examining two alternative filters - Particle Filters and Kalman Filters. The rest of this research paper will be a comparison of the Particle Filters vs Kalman Filters vs the existing solution to see if we are able to successfully track individual baboons through the entirety of the video. Although introducing the usage of filters like that of the Kalman and Particle Filter also brings along a heavy runtime requirement, the current solution fails to track baboons during the entirety of the video. Moreover, the current solution fails to track baboons continuously through lost frames.

Figure 1: Baboons in drone footage

2. Technical Work

We are using continuous, several hour, drone footage to track individual baboons throughout their most active parts of their day. This poses many challenges, logistically and technically. First, we have the issue of working with large amounts of data. Each video is a "4K" resolution which, if not coded correctly, can result in runtimes that exceed the cubic order. Another issue that we must solve is the issue of changing colors. As the footage needs to capture the baboons when they are most active, this includes morning during sunrise. The resulting footage of these times is varied with deep shadows at times vs no shadows and general variation in colors of red, yellow, purple, or blue cast that the sky presents. The group has already created a fairly robust method to track baboons when they are moving, but baboons are highly erratic animals and can often spend long amounts of time sitting in the same spot, wandering slowly, or running rapidly. During these three different states, the "sitting" in the same general area is quite common, and is the vast majority of the actions throughout a baboon's day. Therefore, we are experimenting with particle filters and Kalman Filters in order to attempt to probabilistically predict the actions of baboons at any given state (sitting, walking, or running). In addition, their actions are highly influenced by the time of day, the baboons they are interacting with, and purely their inherent stochastic behaviour. Along with this, there are several trees, shrubs, or rocks that the baboon are occluded behind during the footage. It will be a challenge to follow the baboon behind the occlusions and predict if the same or a different baboon is emerging. This is a particularly difficult challenge due to the erratic movements of the baboons. We cannot assume they will follow a direct path.

2.1. Motion Model

We used a simple baseline fully connected neural network to model the behaviours of a baboon at any given time. In order to train the the network, we need to give it some form of labeled data. In order to have a metric to learn, we are going to base the majority of our predictions on the velocity of the centroid of a baboon. If we can confidently predict the velocity of the baboon from frame to frame, the physical pixel location future belief follows using a simple linear model of distance = (rate) * (time). In all cases, the velocity and location of the baboon is taken in pixelized units. Therefore, it is imperative that we convert all pixels/frame velocities to a unit normalized range since we cannot guarantee the same altitude of the drone for each capture. Once all velocities are normalized to a chosen uniform scale, we can input the velocities into the network to produce a classification. The model was implemented in pytorch, and was initially trained on 53 thousand datapoints, on a Nvidia GTX 1080 Ti. The training algorithm triggered early stopping after the 26th epoch, taking a total of 20 minutes. The architecture of the model is as follows :

- Input Layer : contains an input layer that accepts a vector of size (1 + k + p). The first element represents the current velocity of the baboon in consideration at frame f, the next k elements represents the velocities of the k physically nearest baboons to the baboon in consideration, and the final p elements represents the p last velocities of the baboon in consideration. We have found that not only is the current velocity of the baboon required, but a significant amount of information can be extrapolated from the past velocities of the baboon as well as the movement of baboons around them. It has been empirically observed that baboons congregate together in small packs within the troop in which they belong to. These small packs often make decisions together, and move together and at similar speeds.

- Linear Layer (500, 4096, 25088, 4096, 500) : Each linear layer with respective number of output neurons uses a ReLU activation function along with dropout (0.4).

- Output Layer (output : 5) : The model outputs a softmax probabilistic distribution of which class of movement it predicts. The first class represents the non moving class, the next three represent three classes of walking, and the latter represents the running class.

Each class of walking and running was established using a kmeans clustering algorithm to cluster the velocities of all the baboons during the entire duration of the video. The kmeans model discovers clustered velocities to categorize into different walking and running states. However, it is important to note that the motion model and the kmeans velocity clustering is most effective for full offline (after-the-fact) analysis, rather than an pseudo-online procedure. Furthermore, it also important to make the distinction between pseudo-online and pure offline analysis. We describe pseudo-online as the method in which we take a pre captured video, and run the filters frame by frame without having any knowledge about future velocities or baboons. It is important to do full offline analysis since we need an accurate motion model and kmeans clustering mechanism which needs to know the distribution of velocities throughout the entire video. After the model has "learned" to predict, we can then use this model as the propagation function for the "prediction" step of the Kalman and Particle Filter. The following image represents theh loss convergence that produced an accuracy of 84%.

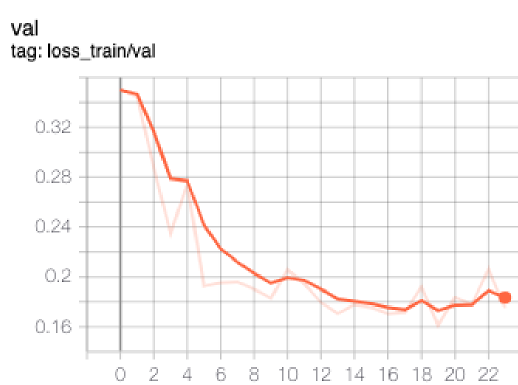

Figure 2: Training Validation Loss on Model

2.2. Particle Filter

The particle filter is crucial to us creating a probabalistic distribution as to where the baboon can be when we lose it from the blob detector for several frames at a time. It helps us to create a heatmap type distribution during the entirety of the video to have an idea as to where that baboon is, since we don't know for 100% certainty where a baboon is if the blob detector for some reason can't localize it. Losing localization from the blob detector not only means we don't where it is, but even worse, when it does reappear, we don't know which baboon centroid is linked to the baboon we were considering. The centroid after the discontinuity can be on the other side of the scene, but it can also be located right where it was before the blob detector lost it. For a high level description of the implementation, we will be having an instance of a Particle Filter for each baboon that we wish to track, and each Particle Filter will contain multiple particles of which are beliefs for that particular baboon's location (with a respective probabilistic weighted confidence). There are three steps to the filtering algorithm that we will go into technical depth : Predict, Update, Resample (and Estimate).

1. Pre-Process step : In this step we must find the nearest baboons that are closest to each of the particles in this particular particle filter. We must iterate through all the other particle filters, and find other baboon's particles that are physically closest to it. This is required by the motion model to predict a viable velocity for the Predict step.
2. Predict : The predict step enters each particle's velocity, their k nearest baboon's velocities, and their p past velocities into the motion model. The motion model outputs a probability for each class, from which we take the product of each probability with the prior probability of the particle before the predict step. We then create (k + p) new particles for each previous particle with these newly calculated weights. After each predict step, we end up with n * (k + p) new particle beliefs (where n is the number of prior particles).
3. Update : If the blob detector can localize the baboon at this given frame, we then alter the weights of the newly created particles. Lowering the probabilities of particles that are physically farther away from the centroid as outputted by the blob detector, and increasing the probabilities of those that are closer.
4. Resample : We eliminate particles that have a low probability, and equalize the weights for the next iteration of the particle filter.

5. Estimate : Here, we estimate the location of the particle filter by taking a weighted average of all the remaining particles. We compute a weighted average for the velocities of all the particles, and compute the location with the previous direction (plus noise), and propagate the previous location with a simple (rate) * (time) calculation. At the end of the estimate step, it is also imperative to record the past p velocities, for later usage in the next predict step's input into the motion model.

We continue this cycle of Predict, Update, Resample, until the video ends, and we can have a probabilistic location for each baboon at each frame of the video which solves the concerns of the previous iteration of this research project.

3. Milestone and Conclusion

We followed the general roadmap of the milestones that we set out in the beginning of the quarter, except for the full completion of the particle filter implementation with the videos. We completed the Kmeans clustering, and training of the motion model on time around week 7, and we completed the bare bones of the particle filter by the end of week 10. Joshua Kang intends to continue work on the research after the completion of the quarter to help finsih the implementation of the particle filter so that it can be used on different videos modularily. Joshua Kang would like to extend the implementation to be largely plug and play given a video so that it can be a seamless pipeline for the researchers. We did not get to finish the full particle filter implementation due to the lack of knowledge from team members, and thus had to spend a week or two reviewing concepts on statistics and probability for the particle filter. The largest problem that arose was getting the particle filter's different functions (Predict, update, and resample) to be written by three different people and put together. I think that this was not anticipated when we were planning out the milestones and is what caused us to digress during the latter end of the quarter.

References

[1] C. Crutcherfield, et al. Baboons on the Move: Enhancing Understanding of Collective Decision Making through Automated Motion Detection from Aerial Drone Footage. 2020