Modeling Movement of Fish

Sudhanshu Agarwal

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science in Computer Science

GEORGIA INSTITUTE OF TECHNOLOGY

Approved by:

Dr. Richard Vuduc
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Jeffrey Young
School of Computer Science
Georgia Institute of Technology

Spring 2022

Abstract

Multi-animal pose detection algorithms such as Social LEAP Estimates Animal Poses (SLEAP) uses a generalized pose prediction algorithm to differentiate among individual animals in the same frame. However, such a method is resource intensive and wasteful when used with animals with simpler movement patterns compared to humans, such as dolphins. My thesis is that the movement of fish such as dolphins can be predicted more efficiently when developing *non-generalized* object tracking methods. This project uses Kalman Filters of size proportional to the size of the "computational skeleton" of a dolphin and is able to accurately classify between different individuals of fish while reducing the size of Kalman Filters by multiple hundred times compared to the method used in SLEAP.

A key feature of my work is the development of non-generalized techniques for animal pose prediction that yield a significant improvement in the speed of pose prediction for fish while preserving the high accuracy of large Kalman Filters. The implications of this result are important as reducing the computational resource load for algorithms such as SLEAP makes them more accessible to a wider set of devices that must operate with fewer resources (e.g., processing cores, memory, energy).

Acknowledgement: This work was supported by the President's Undergraduate Research Award (PURA) for salary awarded by the Georgia Institute of Technology.

Contents

List of Figures

1

Introduction

Tracking animal pose over time is used for understanding neurological and biological aspects of animal behavior, but it is a painstaking and time-consuming process if done manually. In recent years, the development of machine learning based animal pose detection algorithms has significantly sped up the process for quantifying and modeling animal pose.[1] Moreover, computationally analysing animal movement using object detection algorithms has enabled researchers in fields ranging from ecology to neuroscience to develop accurate computational models of animal behavior.[2] The more popular algorithms for pose detection of animals are DeepPoseKit, DeepLabCut, and SLEAP, all of which use convolutional neural networks (CNN).[3] These machine learning techniques make modern animal pose detection algorithms accurate but also resource hungry.

[1] Graving et al. 2019, Singh et al. 2020, Johnson et al. 2019

[2] Johnson et al. 2019, Singh et al. 2020

[3] Graving et al. 2019, Mathis et al. 2018, Pereira et al. 2020

The current state-of-the-art algorithms for pose detection cannot run on resource-constrained devices, such as NVIDIA Jetson Nano, efficiently. For example, DeepPoseKit builds on DeepLabCut and

LEAP,[4] which themselves build on human pose detectors. However, this makes them computationally large because they tend to add excessive neural network layers in order to achieve high accuracy. DeepPoseKit fixes some of these inefficiencies and tries to make inference faster on graphical processing units (GPU) but it still requires significant computational resources. A key reason behind the resource-intensiveness of the current state-of-the-art animal pose detection algorithms is that they are *generalized*, that is, designed to work on any animal. However, they are commonly used in applications that are focused on a specific species or types of animals.[5] This means that developing non-generalized algorithms built on current animal pose detection algorithms, such as SLEAP, can potentially lead to a more efficient algorithm for resource constrained devices. This is important as generalized methods need server-grade hardware, restricting the access to such algorithms for research groups that cannot deploy expensive computing systems.

[4] Graving et al. 2019, Pereira et al. 2019

[5] Johnson et al. 2019

This thesis explores non-generalized methods for quantifying and predicting movement of fish, and thereby, model movement of fish. The software aspect of our project replaces the "animal identification" layer of SLEAP with more efficient solutions and compares the accuracy of the pre-defined methods and the non-generalized ones. The "models" for fish movement are then used to state if there is *uniqueness* to movement of dolphins and if we can *quantify* changes to a dolphin's movement due to environmental changes.

This project's usefulness lies in improving the ability for scientists in other fields to get more data about movement of fish which they can use in their research areas to observe characteristics of fish. From a computer science perspective, this project tries to explore the possible uses of animal pose detection algorithms; one of the use cases is analyzing the data the algorithms generate to find patterns in movement of fish which researchers can use in their research projects. These patterns can further be expanded to uniquely identify fish individuals using just their movement.

2

Literature Review

Computational methods for pose detection of animals have a wide range of applications in neuroscience and biology.[1] However, the modern methods for modeling and analysing pose of animals, such as SLEAP, are built on top of pre-existing tools for human pose estimation with the aim to analyze pose for *all* animals computationally.[2] However, SLEAP is unique as it is able to overcome a key limitation of the animal pose detection algorithms that came before it: it is able to estimate the pose of multiple animals in a single video frame.[3]

[1] Johnson et al. 2019, Singh et al. 2020, Christin et al. 2019

[2] Pereira et al. 2020, Mathis et al. 2018

[3] Pereira et al. 2020, 2019, Mathis et al. 2018

SLEAP is built on the ideas presented in LEAP,[4] which itself uses ideas from DeepLabCut. DeepLabCut[5] uses a variant of DeepCut, an algorithm which uses convolutional neural networks (CNN) to detect human pose,[6] to detect pose of a single animal in a video feed. At a high level, DeepLabCut uses ResNet trained on ImageNet as its backbone for detecting animal objects, and uses transfer learning using user-generated data. Then it combines the data it received from ResNet with deconvolution layers to estimate the position of

[4] Pereira et al. 2020

[5] Mathis et al. 2018

[6] Pishchulin et al. 2016

a body part in a specific location. This is a key difference between DeepLabCut and another animal pose detection algorithm, LEAP, which uses SegNet as its backbone neural network.[7]

[7] Pereira et al. 2019

LEAP uses confidence maps to estimate the movement of a body part of an animal and, in its attempt to reduce model complexity, sacrifices accuracy and is prone to producing erroneous data. DeepLabCut attempts to be more accurate and tends to take more time in inference. Another animal pose algorithm, DeepPoseKit, attempts to fix this trade-off between speed and accuracy by building a new model architecture that they call Stacked DenseNet. The authors of DeepPoseKit introduce a new method of processing confidence maps (for tracking animal body parts) which they call sub-pixel maxima which are optimized for processing on graphical processing units (GPU).[8] While DeepLabCut, DeepPoseKit, and LEAP are different algorithms, there is no *noticeable* difference in their capabilities, and they all lack the ability to track multiple animals in a single frame. Tracking multiple animals is particularly important as this project's goal is use videos of dolphins for testing our algorithm, and dolphins are social creature which tend to live in groups.[9] For these reasons, we use SLEAP in this project due to its ability to work with multiple animals in a single video frame.

[8] Graving et al. 2019

[9] Norris 2002

Multiple animal pose detection requires differentiating between the individuals of animals. SLEAP implements the "differentiating" algorithm in two ways: top-down and bottom-up. Essentially, we can either detect the individuals of fish first and then their pose; or we can detect pose of all animals in a video frame and then classify them as specific individuals. The bottom-up approach in SLEAP of interest for this project as it *builds on top* of work done in LEAP, and so we can perform comparative analysis of speed and efficiency of adding multiple animal support. The bottom-up approach uses Kalman Filters to classify different individuals of animals. Essentially, SLEAP attempts to predict the *most likely* position of the animals in the next video frame and than does a closest location match with the pose it generated for all animals in the said frame.

The two methods used for predicting movement of animals are using the flow algorithm adapted from those for human pose prediction[10] and using Kalman Filters. The latter is of interest for this project as Kalman Filters are self-correcting models which estimate the relationship between the variables fed into the algorithm.[11] Kalman Filters use measurements recorded over time, such as the movement of the computational skeleton of the fish to predict where the fish is going to move. This filter has been studied and applied extensively to fields such as automation and navigation, but it can also be applied to predicting and developing models of how pose affects the movement of fish in water and can further involve variables like the flow of water. This is useful for this project as the model generated by running Kalman Filters is a potential method for understanding if movement of fish can be uniquely identified.[12]

[10] Xiao et al. 2018

[11] Bishop et al. 2001

[12] If we are able to create accurate prediction models for movement of an individual of fish, then these models can be compared with those for other individuals to see if each individual has a unique predictive model associated with it. This can imply that there is a unique predictive model with which it can be identified.

3

Methodology

The test bed for the experimental evaluation of your algorithm was a single NVIDIA Jetson Nano with Ubuntu 18.04. The NVIDIA Jetson Nano has a quad-core 1.43 GHz ARM A57 processor with a 128-core Maxwell GPU.[1] However, SLEAP was trained on Google Colab.

[1] NVIDIA. Jetson Nano, March 2019. URL https://developer.nvidia.com/embedded/jetson-nano

3.1 Training SLEAP

In this thesis, the movement and pose of fish is detected using SLEAP as it is the current state-of-the-art for generalized multi-animal pose detection algorithms.[2] SLEAP uses a “computational skeleton” of animals of interest to train its neural network to train neural networks for pose detection. For example, initially we use a five node skeletal system with a “dense” and “linear” bone skeleton that represents two potential matrices that correspond to the assumed skeletal structure of dolphins (see fig. 3.1). The computational skeleton is primarily used for training SLEAP, so our skeletal structure for dolphins can be significantly smaller (and simpler) than an actual dolphin skele-

[2] Pereira et al. 2020

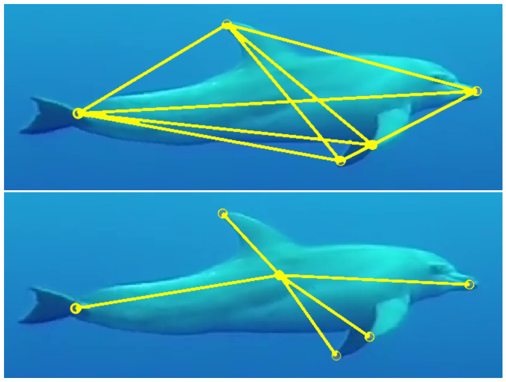

Figure 3.1: An example of a "dense' (top) and "sparse" (bottom) skeletal structure for dolphins used for training SLEAP.[3]

ton. However, this smaller skeleton should be able to model *most* of the movement of a dolphin.

[3] This is a grab from a video feed used for training and testing the algorithm developed in this thesis.[Soothing Relaxation]

3.2 *Predicting Pose Data*

SLEAP performs inference in two steps in the "top-down" approach: pose detection and animal identification. The first step is left unmodified in our analysis as that is *shown* to be state-of-the-art.[4] The second part of the algorithm is of more interest to us as we are aiming to replace the generalized Kalman Filter in this part with a non-generalized one.

[4] Pereira et al. 2020

Here, we use a Kalman Filter proportional to the number of nodes (or vertices) in the "computational skeleton" where the x and y axis of vertices are passed as a single linear buffer. This Kalman Filter is then used to generate predictions for future movement of dolphins in the video feed, and these predictions are tested against the actual movement of the fish to see if they are accurate. If the accuracy is not high enough, then we will create a Kalman Filter with more parameters and a more complex "computational skeleton" of fish. However, a large Kalman Filter does lead to an increase in the computational resource requirements for our project if we want to perform predictions in real-time.

To compare our method with the one built into SLEAP, we analyze the efficiency and accuracy of our prediction layer. Efficiency can be compared by the size of Kalman Filter matrices (smaller is better) and accuracy by manually comparing the predictions for the next frame with those generated by SLEAP and the actual movement of dolphins.

3.3 *Uniqueness of movement*

If we are able to generate a Kalman Filter that is *accurately* able to predict the future pose of an individual dolphin, then we can assume that Kalman Filter is *associated* with that specific individual. However, we need to show that there is a unique Kalman Filter associated with all individuals of dolphins. Just as importantly, given any individual of dolphin, we can generate a *unique* pose prediction model for it. This idea can be argued using mathematical theory or by trial-and-error. We plan on pursuing the latter approach as this thesis is more focused on an application of Kalman Filters, rather than their mathematical properties.

Note: All code is available on Github.[5] The dataset was built using [Soothing Relaxation] video on YouTube and please contact the author for access to it.

[5] Sudhanshu Agarwal. Modeling movement of fish, 2022. URL https://github.com/sudhanshu2/modeling-fish-movement

4

Results and Discussion

4.1 Generating an accurate "computational skeleton" for dolphins

The initial "computational skeleton" used to training SLEAP was a four node "dense" structure shown in fig. 3.1. This skeleton was used to develop a training dataset with approximately 100 images, which is relatively low for a generalization. Nevertheless, we were able to train SLEAP with an high accuracy (see fig. 4.1) and with low overfitting. This accuracy can potentially be increased by adding more nodes to the "computational skeleton" of the dolphins or by increasing the number of images. However, in order to keep the size of the prediction layer to the minimum, we did not increase the number of nodes in the computational skeleton. Testing with more images did not yield a significant improvement to the training accuracy of SLEAP. Thus, we were able to infer that our current dataset was able to preserve the aspects of dolphin motion while being efficient and accurate.

Figure 4.1: The heatmap shows the estimations for the centroid of the computational skeleton of the dolphin. The estimations were very accurate even with a relatively small training data set.

The trained SLEAP model was then used for generating pose data for a different video feed with two dolphins swimming close to each other. This is a particularly hard problem as the individuals of dolphins are very close to each other and a "predictive" layer of pose can misidentify these individuals if the prediction is not accurate enough. This misidentification did occur multiple times in our testing video feed with the built-in Kalman Filter in SLEAP. However, the model sampled enough data during to generate correct results in inference.

4.2 *Kalman Filter*

In this step, we attempt to override the pose prediction layer of SLEAP by testing a smaller (and more efficient) Kalman Filter model. The coordinates of the current pose of a dolphin are passed to a unique Unscented Kalman Filter[1] (part of the `pykalman` Python package) related to each individual of dolphin. This Kalman Filter then produces probable predictions and "corrects" itself over time as it gets more data. The predictions generated by the Kalman Filter were very inaccurate in the beginning but improved in the next 40 frames of predictions. This means that the movement of pose of dolphins is *learnable* by a single Kalman Filter. More importantly, we are able to predict the pose of dolphins using just four nodes instead of replicating the actual skeletal structure of dolphins, which consists of over 200 bones.[2] Just as importantly, the Kalman Filter size used in this project is smaller than that used by SLEAP by default along with fewer computations on the Kalman Filter to estimate the future pose.

[1] Wan and Van Der Merwe 2000

[2] Cozzi et al. 2016

Moreover, the ability to accurately model the pose of dolphins means that we can analyze the generated Kalman Filter to determine if it is unique for each individual of fish. The analysis of the model will depend on the `transitions_functions` and `observation_functions` in `pykalman` as these are used for next frame predictions. We found in our testing that these functions were indeed unique for different individuals of dolphins. However, these differences do not mean that the movement of different individuals of dolphins are different. The Kalman Filter was run on a video feed. and that means if we change the video feed (for example, by cropping it), the filter for the same individual of dolphin will change. But, we were able to show that if we use a "fixed" viewing angle for different individuals of dolphins, we should get unique Kalman Filters for all individuals without such a Filter being a "universal" predictor for pose for a dolphin.

5

Conclusion and Future Work

Prediction and modeling of the movement of animals should focus on developing non-generalized techniques because there are field-specific assumptions and efficiencies that can be exploited when developing these methods to reduce model size and complexity.

I focused this work on dolphins, but the results and software architecture developed in this thesis can be adapted to other animals with "fish-like" movement. This means that this algorithm encompasses the whole research area of fish research in marine biology. More importantly, using algorithms that reduce the computational intensity for algorithms can reduce the greenhouse emissions for running powerful computers that algorithms such as SLEAP might require.

The ability to train an accurate model for prediction of pose for dolphins enables us to perform comparative analysis of changes in movement of dolphins by changing environmental conditions. Essentially, we can generate a pose prediction model before and after the change in environmental effects and can analyze *how* the model changes. Moreover, if we are able to make sure that the video feed used for pose prediction always has the same "fixed" view of the dolphins, then performing comparative analysis at a large scale can lead to significant improvement in our ability to *quantify* climate change's effects on marine life.

The results of this thesis show that a "computational skeleton" does not have to be representative of the actual anatomy of dolphins in order to generate accurate pose prediction models using Unscented Kalman Filters. This finding enables us to potentially *reuse* computational skeletons as the one used in this thesis can also be adapted for other fish such as whales and sharks. Doing so reduces the time it takes to develop training data as we can automate the ask of assigning computational skeletons to fish objects in a dataset.

There is significant future work in this field, especially when it comes to the uniqueness to movement of fish. While we were able to show that all fish in our testing had unique movement characteristics, we cannot generalize this result to *all* dolphins. Further, we should attempt to use other non-linear movement modelling techniques other than Kalman Filters to see if the prediction of pose of fish can be made even more efficient. Just as importantly, there is work to be done in developing non-generalized pose prediction algorithms for other types of animals such as elephants or reptiles.

6

Bibliography

Sudhanshu Agarwal. Modeling movement of fish, 2022. URL https://github.com/sudhanshu2/modeling-fish-movement.

Gary Bishop, Greg Welch, et al. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

Sylvain Christin, Éric Hervet, and Nicolas Lecomte. Applications for deep learning in ecology. *Methods in Ecology and Evolution*, 10(10): 1632–1644, 2019.

Bruno Cozzi, Stefan Huggenberger, and Helmut A Oelschläger. *Anatomy of dolphins: insights into body structure and function*. Academic Press, 2016.

Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife*, 8:e47994, 2019.

Robert Evan Johnson, Scott Linderman, Thomas Panier, Caroline Lei Wee, Erin Song, Kristian Joseph Herrera, Andrew Miller, and Florian Engert. Probabilistic models of larval zebrafish behavior: structure on many scales. *bioRxiv*, page 672246, 2019.

Alexander Mathis, Pranav Mamidanna, Taiga Abe, Kevin M Cury, Venkatesh N Murthy, Mackenzie W Mathis, and Matthias Bethge. Markerless tracking of user-defined features with deep learning. *arXiv preprint arXiv:1804.03142*, 2018.

Scott Norris. Creatures of culture? making the case for cultural systems in whales and dolphins. *Bioscience*, 52(1):9–14, 2002.

NVIDIA. Jetson Nano, March 2019. URL https://developer.nvidia.com/embedded/jetson-nano.

Talmo D Pereira, Diego E Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S-H Wang, Mala Murthy, and Joshua W Shaevitz. Fast animal pose estimation using deep neural networks. *Nature methods*, 16(1):117–125, 2019.

Talmo D Pereira, Nathaniel Tabris, Junyu Li, Shruthi Ravindranath, Eleni S Papadoyannis, Z Yan Wang, David M Turner, Grace McKenzie-Smith, Sarah D Kocher, Annegret L Falkner, et al. Sleap: multi-animal pose tracking. *BioRxiv*, 2020.

Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

Satpreet H Singh, Steven M Peterson, Rajesh PN Rao, and Bingni W Brunton. Investigating naturalistic hand movements by behavior mining in long-term video and neural recordings. *arXiv preprint arXiv:2001.08349*, 2020.

Soothing Relaxation. 8 hours of relaxing sleep music: Deep sleeping music, calming music, relaxing music. URL https://www.youtube.com/watch?v=1UcozXB5Pg0.

Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.