BIOL792, YYYY, 1–3
doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

Predictive Image Modeling from LiDaR Data

Arturo Macias Franco

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Though predictive image modeling is not something new for animal science, its applications to large-scale systems is still an emphasis of current research endeavors. Through my PhD work, I aim to incorporate technologies available for different industries and develop frameworks to incorporate them to increase sustainability of agricultural operations. Primarily, a large component of my work involves livestock nutriton. Specifically, I work with nutrient requirements of bovine species in rangeland and intesive feedlot operations. Through our research, we aim to address biological dilemmas, and generally, agricultural issues through the use of Mathematical models. More formally, I generally try to formulate predictions that allow us to better understand physiological processes related to production such as feed and water efficiency, environmental footprints, predicted feed and water intakes, prediction of incidences of particular disease such as bloat, amongst others. More recently, a side-project we took on involved the use of LiDaR cameras from intel to generate predictions we have already done from morphometric measures which will be the focus of this project.

Key words: LiDaR; mathematical modeling; body composition

1. Introduction

For this class project, I was originally interested in automating data extraction from LiDaR images and videos. I was originally given some code from some peers with the impression that they had a pipeline for extracting the data from images. From conversation, they had no idea why the code was not working, for it had been written by an engineer in their school. I said I would troubleshoot it, and simply establish a bash script where I could apply it to the data files I had and would be able to present on this pipeline and do some modeling.

Unfortunately, upon receiving the code and beginning to work on it, a large component of my time was spent on translating the code and making sense on what the 15+ python files were doing. I was finally able to get the code running with the help of pyCharm which mostly helped me detect typos, and issues with the structure of the code.

Ultimately, I was surprised to find out that the code was put together with the purpose of compiling a desktop application to manually capture and import images for analysis which completely threw me off of my tract. Further, upon compiling this document, I already started thinking about ideas on how I can go about automating this extraction and is something I will try to spend my winter-break on.

Nonetheless, this was a rewarding experience in that I was able to work on some coding skills not necessarily related to statistical analysis, but rather, for the development of an app. Further, compiling of this code made it necessary for me to get more comfortable with pycharm which has already proven a rewarding endeavor within itself.

2. Framework for running the code in pycharm

1) download the zip files and establish them in a directory where you will want to run the project from. During the data analysis, there is an option to extract the data, thus, having this organized since the beginning will help you access this data more easily.
2) Once the file has been unzipped, we launch pycharm, either through anaconda, or separately installed. It should notify you which of the packages you may be missing for the code to run. In establishing the directory

Inside of the zip file:

3) The analysis file is likely the most important folder, inside of the Analyzer.py, we have the code that will actually perform all of the computation we are interested in. This is where I will be looking to automate this process without having to launch the interface. **WORK IN PROGRESS**
4) The Capture folder delines the code followed for detecting of the LiDaR camera, if you guys are interested in running this code, I could let you borrow one of the cameras I have here at the lab, but otherwise, it is just there to detect the cameras, and has a flowchart of options through which you can assign specific features (like turning on the depth imaging on top of the RGB imaging on display), there's also neat features for incorporating noise reduction.
5) On the Data, the folder Boi will have some template images from some animals we scanned, feel free to load these images during the analysis, there will be options to add an image (where we will add the color image), and an option to add the depth image (where we will add the depth).
6) Lastly, the folder labeled Interface is the individual code files utilized for the compiling of the desktop application, I tried to add some tips on the app itself that should allow you guys to run it. It is fairly intuitive.
7) The RSM is where the magic will now happen to bring everything together for capturing imaging, loading images, getting data, and the main.py file is the compiler we will run from pycharm to build the desktop application. *Notice for those inexperienced with pycharm as I was, to run a file in pycharm, you need to first have the directory established. Once that is set on the file containing the zipped archive, you will have to go to the option Run, which will ask which code you want to run. In order to compile the application, we run the main.py file which takes around 20 seconds to launch the desktop application for you guys to start working*
8) The first thing you will see is a little window open saying capture image, or image analysis, if you have an intel realsense camera, you can capture an image, otherwise you will need to press on image analysis. Once you press image analysis, it will open a new tab for you.
9) Once you are in the new tab, read over the instructions on how to use the program, it should be fairly intuitive. The first thing we will want to do is go to File on the top left, and select load data/values and click on the option new tab. We can have several tabs open with many different images.
10) Once the new tab is open, notice that a features menu appeared on the right hand side. We are now ready to start the image analysis and data extraction.
11) We will now load color and depth images for the analysis, since we are trying to estimate body volume. We will again go to the select load data/values under the top left File button, we will load a front color image and a front depth image. I left image_color1_n89 on the main menu to make this easier, an image should pop up for you guys now.
12) Once the image is loaded, we need to establish where we want to compute the estimation for area, volume, and perimeter, for this, we use the buttons on the right hand of the screen where we will define the limits of the image. We need to cut the box where we will want the computation made. *we generally exclude the head and just box in the main body. Also notice, when making the cuts, you need to click on select and then choose the area of the image that you want to box in*
13) Once we determined the box where we will want to extract the features from, we need to add the distance from the camera to the steer. By default, we always have the image placed 3 meters above ground.
14) There are some additional tuning parameters for depth, horizontal cut, and vertical cut that can be adjusted to get a better representation/cut, play with them to see how it changes the estimated areas **ONLY AFTER PROCESSING FOR THE FIRST TIME**.
15) We now press Process, look on the bottom "console" it will print out the data that we want to print out.
16) The obtain descriptors button opens a picture detailing the different cuts utilized for estimation, they offer insights into the data that is offered when you **SAVE DATA**.
17) Some additional features are the split to add splits to the image to assist with analysis, and the main core analysis which converts the cut image into black and white.

3. Conclusion

Though my original pursuit on this project was overly-ambitious, I am content with the progress I made in being able to comfortable work from the terminal and on the continuous adaptation I am finding on incorporating python to my research. Ultimately, with this work, I could, in theory, process images one at a time to extract the data for my predictions. Unfortunately for me, I am not patient enough, and will likely spend more time developing the pipeline to automate this instead of processing the images one by one and likely finishing this sooner than I would trying to develop the code.

As far as future steps, I will be working on the Analyzer.py code with hopes of automating the extraction process without the need of a user interface. Some initial ideas I have for that involve complete extraction with all data. For the purpose of this data, if the error/noise was the same for all images, I could try to detect patterns that would allow me to estimate from the image. Additionally, I found some code that I could incorporate that would allow me to incorporate a filter that removes data according to distance from the laser scanner, given that the distance is always the same, determining a measure of centrality for height that would allow me to clear a lot of the noise on the images could allow the automation of this step.

I will be attempting to spend some time during winter break working with some faculty on my department that work with some of these technologies specifically applied to rangeland/forestry settings with hopes that I can pick up some tools that will assist me in my endeavors.

Arturo Macias Franco A promising young researcher working on his thesis project. This was quite hilarious from the template, so I am keeping it.