Lab 05 - Tracking the jump of the trap-jaw ant

How Organisms Move, Bio 490S, 2014

1 Goals

The ability to convert digital images that capture data from the world is an important skill, and given the ubiquity of camera technology in the modern world this skill is increasingly useful. By understanding how to process digital images into manageable datasets, researchers have created algorithms which can recognize individuals from security camera footage as well as those which can identify and track stars using images from telescopes. The ability to process image data is particularly applicable in the study of extremely fast biological movement, where high speed video can capture one second of life in over 1,000,000 frames.

Although these movements involve elastic mechanisms and occur on extremely fast time scales, we can use some basic tools and equations to understand and analyze them. This lab will make use of video recorded from a high speed camera to analyze the ballistic trajectory of *Odontomachus bauri*, the trap-jaw ant. We will be using imageJ, an image program that researchers often use to make numerical measurements from images and videos, to manually collect data from a trap-jaw ant during its jump, and then use R to calculate velocity, take-off angle, and projected distance of the jump. We'll then take a computer-vision approach to automate these calculations. Finally, you'll have the opportunity to use these tools and techniques to conduct a small independent project using videos from the internet to answer a question of your own.

1.1 What is due

This weekâĂŹs assignment is listed at the end of the lab manual. Type your response into a Word document and upload it to Sakai before the due date (11 pm Sunday). Include your name, date, and lab title at the top of the document.

2 New techniques

2.1 Installing imageJ

ImageJ is free software developed at the NIH. This program is helpful for tracking movement in a series of frames of a video. Download Image at `http://imagej.nih.gov/ij/download.html`.

2.2 Opening files in imageJ

The high speed video is collected as group of still frames which we will call an image sequence. To import the image sequence into imageJ, click file > import > image sequence. Navigate to the folder containing the images, select the first image in the sequence and click "OK". A new window will pop up called "Sequence Options". Although in practice, we would like to deal with as much data as possible, for the purposes of this lab, we will limit ourself to 100 frames. Additionally, the first few frames of our video may not contain the trap-jaw ant jump, so we will start with image 20. Finally, the high speed video captured the ant jump with 3000 frames per second, but for this lab, this data acquisition rate is higher than it needs to be. We can tell imageJ to skip frames by changing the Increment from 1 to 2. This will effectively change

the fps from 3000 to 1500 fps. After inputting these numbers, check your video. You can scroll through the video using the scroll wheel of your mouse. Check to make sure that by the first frame, the trap-jaw ant is already in the air moving. If not, try reimporting the image stack using a higher value for the start image.

2.3 Setting the scale

In all of these videos, there is a scale bar in the background that provides us with information that we can us to convert points in the video into real distances. We will use the magnifying glass tool  to zoom in on the scale bar. Once this tool is selected, left-click to zoom in and right-click to zoom out. Once the scale bar fills the entire view of the screen, we can draw a line onto the scale bar using the line tool  . With this tool selected, drag a line from one marking on the scale bar to another. For simplicity, it will be easier if you draw your line to represent the distance of 1 cm. Finally, click analyze, input the known distance of 1.00 and set the unit of length to cm.

2.4 Tracking points

To track the position of the head, we will need to change from the single point tool  to the multipoint tool  . To do this, right-click on the single point tool and select multi-point. Scroll through the video to make sure that we are at the beginning, zoom in on the ant, and choose a point on the head that is easy to track throughout the ant's jump. Click on this region to add a data point and notice that the point is numbered. If you want to move the point around, you can drag it on the screen. Keep in mind that the more accurately you track the data points, the more realistic your calculations will be. We recommend zooming in before adding data points.Once you are satisfied with the position of the point, scroll to the next frame of the video and continue to add points to track the ant's head through the video. Make sure that when you click, you are actually adding a point. Write down the total number of points tracked since this will be important in the future.

2.5 Exporting data

To save the data of our tracked points, click analyze > measure. This will open a new window with a lot of information about our points including the x and y positions. Click file > save as and make sure to name your file with a *.txt extension.

3 Lab

This week's lab consists of three parts: 1) manually tracking the trap-jaw ant, 2) automatically tracking the trap-jaw ant, and 3) a small independent project.

3.1 Part 1: Manually tracking data points

Using the video of trap-jaw ant jumping vertically, create two *.txt files from imageJ (one with the tracking information for the head and another for the abdomen). You will need to create these *.txt files separately, and make sure that both datasets have the same number of tracked points for this lab to work. We will use these two datasets to calculate and plot the trajectory of the ant body, and then apply physics equations to see if the ant's trajectory can be predicted with ballistic equations. Below is a guide as to what you need to do to complete this lab:

1. Download antJump.zip to get all the frames of the video. Once you download the *.zip, double-click this file and drag the folder to your hard drive.

2. Ensure that all frames being analyzed are of the ant in mid-jump. Make sure to write down the numbers you use in the "sequence options" box in case you want to reimport the video at a later time.
3. Set the scale of the video.
4. Track the head and export the data.
5. Track the abdomen and export the data.
6. Make sure the two datasets are of the same length (both track the same number of points).
7. Import both datasets into R.
8. Download and source the script plotAnt.r from Sakai.
9. Use the plotAnt() function to plot the trajectory of the ant. This function takes four inputs: the x values for the head, the y values for the head, the x vales for the abdomen and the y values for the abdomen. Inspect your datasets in R using the str() function to find the names of your columns of x and y positions. If you are unsure of what the columns are referring to, don't hesitate to open the *.txt files in notepad to investigate. Also, use the dollar sign ($) to get the x values from your dataset. For example, if your dataset is named antHead, you can retrieve the x and y values by typing antHead$X and antHead$Y. After running the plotAnt() function, you will see an animation of the ant jump. The blue line shows the trajectory of the head, the red line shows the trajectory of the abdomen, and the black line is the calculated path of the ant's body.
10. The plotAnt() function also outputs the x and y coordinates used to create the path of the ant's body. Store this information into a variable in R called antPath.

3.2 Part 2: Automated tracking

For the automated tracking, we will use a modified dataset of the trap-jaw ant jump. Before this lab, we used a computer vision algorithm to subtract the background from the video, similar to the special effects green screen technique. We then used this information to set all pixels of the background to black and all pixels that were the ant to white. The following exercise will show you how data can be collected from images automatically.

1. Download antJumpAutomatic.zip to get all the frames of the modified video. Once you download the *.zip, double-click this file and drag the folder to your hard drive. Open a few of the images to see what they look like.
2. Download and source the script automaticAnt.R from Sakai.
3. Install the "jpeg" package:

```
install.packages("jpeg")
```

4. Load the "jpeg" package:

```
library(jpeg)
```

5. Set your working directory to the folder unzipped from antJumpAutomatic.zip
6. Import the first frame of the video and inspect the output:

```
readJPEG("ant_0001.jpeg") -> ant1
ant1
```

7. ant1 is a matrix where each cell represents a single pixel of the imported image. Black pixels are converted to 0s and white pixels are converted to 1s. We're only concerned with the ant (white pixels), so convert the matrix into a list of x,y coordinates of all the white pixels:

```
imageToXY(image=ant1, plot=T) -> ant1.xy
ant1.xy
```

8. Find the center of the ant in this frame. A good estimate of the location of the ant's center is the mean of all the ant's pixels:

```
mean(ant1.xy$x)
mean(ant1.xy$y)
```

9. Repeat steps 6 - 8 for frames 100, 400, and 700.

10. Instead of doing these calculations for all 751 frames, we've provided a function to automate all the calculations for you. Run the script to automatically process through all the data:

```
automateAnt() -> antPath-auto
```

3.3 Part 3: Independent project

Now that you've learned the basic skills behind converting video to data, you'll use those techniques to analyze kinematic video of your choice. Think of an interesting question about movement that you can use videos to answer. Then, go out and get those videos either by recording them yourself or by searching on youtube. Once you find or record videos that will help you answer your question, we'll help you convert them into a format that imageJ can import.

4 Turn in

1. Include the two plots (using ImageJ and using the automatic script) of your ant jump with figure captions.

2. Typing in antPath, the variable created from step 10 in part 1 of the lab, you will notice that R will return the x and y coordinates of the path created by the ant's body. Using the first few values in antPath and the frame rate of the video, calculate the initial velocity of the ant as well as the angle of the trajectory. Do the same calculations using antPath-auto, which was created using the automatic program.

3. Using the initial velocity and angle of trajectory from antPath, what is the projected horizontal distance of the ant in your jump. Does the projected horizontal distance change when using data from antPath-auto? Does your calculated distance agree with how far the ant actually lands from the initial point? Make sure to include any equations you used in your answer.

4. What are the major benefits of using an automatic tracking system to collecting data based on this lab and on the literature provided? What are the major drawbacks?

5. Provide a single paragraph describing what you did in part 3 of the lab. What was the question you were trying to answer, and based on your analysis, what is the answer? If you could collect more data (i.e., capture more video) what steps would you take to make the analysis either easier or more robust?

You may find this study of ant jumping and trajectories helpful as you perform your own analysis:

Patek, S. N., Baio, J. E., Fisher, B. F. and Suarez, A. V. (2006). Multifunctionality and mechanical origins: ballistic jaw propulsion in trap-jaw ants. Proceedings of the National Academy of Sciences 103, 12787-12792.

And, you may enjoy these youtube videos of the jumps:

`http://www.youtube.com/watch?v=G89IcZ3PluE`

`http://www.youtube.com/user/pateklab?feature=results_main`