Quantitative Measurements of Forward Flapping Flight Using Image Processing

Eric Gutierrez
Aeronautics and Astronautics
Stanford University
Stanford, CA 94305
eguti007@stanford.edu

***Abstract*—A method for automatically calculating the flight kinematics of a flapping bird flying on a relatively dense background environment was developed. The algorithm is able to calculate the bird's flapping frequency, wingbeat phase, forward velocity, and forward acceleration with good accuracy with the use of only one camera. Image processing was used for extracting the bird's silhouette from the background. An image matching algorithm, which made use of the Kullback–Leibler divergence, was used for matching a unique template frame to similar frames of the flight. The algorithm was tested on three full flights of one bird. The algorithm was able to detect similar wingbeat phases with 100% detection accuracy for every data set, giving good accuracy for the flight kinematics.**

Keywords—image processing, KL divergence, bird flight, kinematics

I. INTRODUCTION

In bird flight research, it is important to quantitatively and accurately estimate the flapping flight kinematics of the bird being studied. For instance, for studying the wake aerodynamics of a bird using particle image velocimetry techniques, an accurate representation of the wingbeat kinematics of the bird is important for accurately estimating the flight forces. Different flight kinematics include the wingbeat frequency, wingbeat amplitude, wing phase, and flight velocity [1]. Knowing if the bird is accelerating, decelerating, climbing, or descending during the measurements, or data acquisition, is also important.

It is common for researchers to paint or glue markers on the wing and body of the birds for obtaining flight kinematics [2, 3]. In these studies, two or more cameras are usually used for obtaining accurate measurements. Since this sometimes requires the researcher to manually digitize the data by clicking the markers on a computer, this method for obtaining flight kinematics can be time consuming. It is also stressful for the bird since the bird needs to be held in place while the markers are put on the wings and body. This paper explains how to obtain flight kinematics using image processing and image matching techniques with only one camera.

II. EXPERIMENTAL SETUP

The evolution of the wake vortices from a freely flying Pacific Parrotlet (*Forpus coelestis*) was examined in still air. The bird was trained to fly 1.5 meters from one perch to another perch in an approximately straight path through a laser sheet while wearing custom-made laser safety goggles. This enabled a detailed study of the evolution of the vortices shed in its wake using stereo particle image velocimetry. A similar study which utilizes PIV for studying bird flight can be found in reference [1]. Four high-speed cameras (Phantom Miro M310) recording at 1,000 frames per second were used to study the motion of the air particles left behind the bird. A fifth high speed camera (Phantom Miro LC310) also recording at 1000 frames per second was used for obtaining flight kinematics of the bird. The kinematics camera was positioned behind the take-off perch to track the bird's entire flight path (Fig. 1).

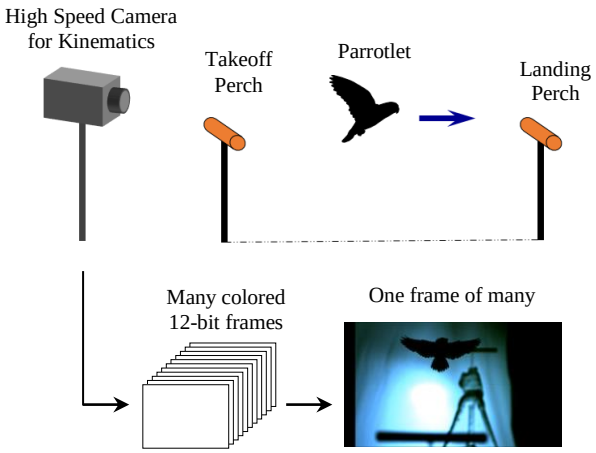

Fig. 1. Setup of the experiment for obtaining the entire flight of the bird.

With positive reinforment, the bird was trained to fly to the landing perch whenever I pointed a finger at the perch. After the flight, the bird was rewarded with a millet seed. Five succsefull flights were recorded. For two of the flights, the bird was out of the field of view of the kinematics camera for some frames. Therefore, for consistency in the data set between the flights, three flights were chosen and analyzed. Each flight contained approximately 700 frames and 18 full wingbeats from the bird.

III. METHODS

The main goal of the study was to accurately and efficiently calculate the flight kinematics of the bird with one camera. Two separate algorithms were written in MATLAB to accomplish this task. Each will be described in separate sections. The first is titled "Image Processing: Bird and Background Isolation", and the second is titled "Image Matching."

A. Image Processing: Bird and Background Isolation

Before calculating the flight kinematics, I had to extract the bird's silhouette from the background as accurately as possible in order to obtain accurate estimates of the flapping frequency, wingbeat phase, forward velocity, and forward acceleration. For images where the background can be easily distinguished from the foreground, this can be accomplished relatively easily by performing image thresholding with algorithms such as Otsu's method. However, the frames acquired from the kinematics camera contain the bird, the perches, a perch stand, and a heterogeneously illuminated background. In some frames, the landing perch interferes with the bird's wing.

Fig. 2 contains images of the image processing steps for accurately extracting the bird's silhouette from the cluttered background. The steps are as follows:

a. Import every frame (12-bit colored images) into MATLAB. Each frame is 1920 by 1080 pixels. To decrease the processing time, every frame is converted to grayscale and rescaled downwards by 1/2. In addtion, a smaller field of view where the bird appears in every frame is chosen for analysis.

b. Calcuate the background frame without the bird by using two frames where the bird appears in two different locations.

c. Blur the background image with a gaussian filter of size sqrt(20) pixels.

d. Blur every frame with a gaussian filter of size sqrt(20) pixels. Since gas particles micrometers in size are being illuminated by a laser sheet while the bird is flying, the brightness of pixels are constantly changing for every frame. Blurring the image will reduce these fine detail changes, which will result in the blurred background image (step B) being more similar to the background of each frame. This will give better results when eliminating the background from each frame.

e. Eliminate the background from each frame by comparing the pixels from the blurred background image (step C) to the blurred frames (step D). If the difference between the blurred background image and the blurred frames is greater than 15 gray level (pixel by pixel), then this is estimated as part of the bird. If the diference is less than 15 gray levels, this is estimated as part of the background. This number was well chosen or else too much noise would be introduced or parts of the bird would be erased.

f. Threshold each frame with a threshold value of 0.3. This threshold needs to be large enough in order to not eliminate the light parts of the bird and low enough in order to not introduce too much noise from the background. Here, objects not attached to the bird were eliminated by using MATLAB's bwareopen function.

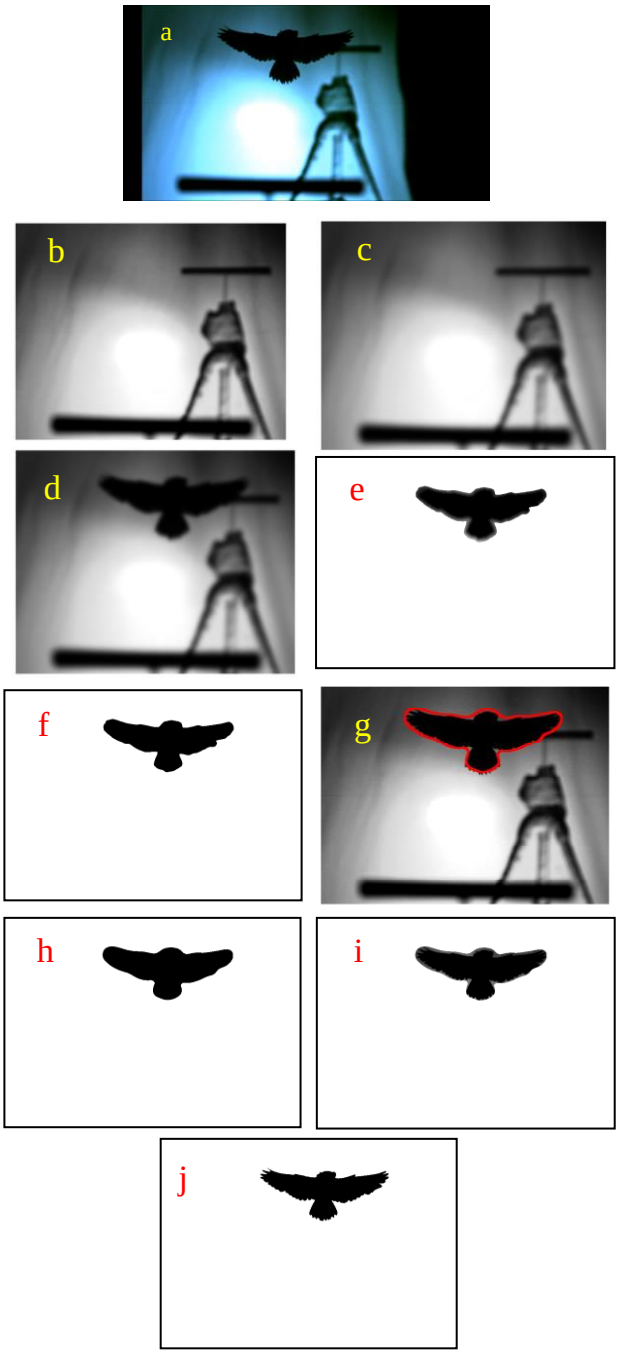

Fig. 2. Image processing steps for accurately obtaining binary frames of the bird's silhouette for the entire flight path. The final binary image of the silhoutette is shown in J.

g. To smooth the outline of the bird, the canny edge detector with a gausian filter of sqrt(80) was applied. For some frames, parts of the background (e.g. black perch) were wrongly considered as part of the right wing of the bird. For other frames, segments of the bird's right wing were

erased due to the perch being a similar gray level to bird. This resulted in erroneous peaks extruding outward or inward from the bird. A comparison before and after applying the canny edge detector is shown in E and I. Sometimes, the canny edge detector did not give closed outlines. Therefore, the canny edge was dilated until the outline became closed. The Laplacian of Gaussian edge detector was also tested; this edge detector did not smooth the outline, no matter the size of the gaussian chosen.

h. The binarized frame after applying the canny edge detector. Only the pixels inside the canny edge are shown in black.

i. Improved image E after applying the canny edge detector. Only the pixels inside the canny edge are chosen for analysis. It can clearly be seen that the outline of the right wing is improved. Part of the right wing was cut off in image E but not in image I.

j. Thresholded the frames with a threshold value of 0.15. The fine details of the bird (e.g. feathers) can clearly be seen. The silhoulette of the bird for each frame was extracted similiarly to this particular frame.

B. Image Matching

The binary images obtained in step J of the image processing section were used for calculating the flight kinematics of the bird. The most trivial flight parameter that can be calculated with image processing is the bird's flapping frequency. Flapping frequency as a function of frame number can be calculated by finding when a wingbeat repeats itself. An algorithm that automatically detects and matches the frames where the position of the bird's wings are most similar to a template frame would be very useful for calculating the flapping frequency.

First, I used the Scale Invariant Feature Transform (SIFT) algorithm, an image descriptor for image-based matching and recognition [4, 5]. SIFT is widely used for matching points between images. I found that SIFT worked very poorly for all three videos. The reason is because SIFT describes keypoints by using gradients in the image. Since the bird itself didn't have enough texture on its surface and was mostly covered with zero-value gray level pixels, only a few descriptors (about three) were computed on the surface of the bird for every frame. These were not enough descriptors for accurately matching frames.

The second image matching algorithm I tested was the Kullback-Leibler divergence method. This algorithm worked almost perfectly for matching frames. The steps for this algorithm are as follows:

1. Calculate the outline of the bird from the binary frames obtained in step J of the image processing section. The oultine needs to be 1 pixel thick.
2. Calculate the centroid of the bird for each frame. A filter needs to be applied to the centroid indices to damp the high frequency oscilation of the centroid due to the flapping of the wings. I applied a hann filter to my algorithm.
3. Calculate the distance between the centroid of the bird and each pixel that pertains to the bird's outline. Normalize each vector by the median distance of each frame to take care of the change in size of the bird.
4. Create a probability distribution of the distances.
5. Compute the Kullback-Leibler divergence (KL divergence) (difference between probability distributions) between a template frame and every frame of the flight. Fig. 3 shows the KL divergence for the first flight. All three flights showed similar KL divergence behavior. The equation for the KL divergence used in this study is as follows:

$$D_{KL} = \sum_i (P(i) - Q(i)) \, ln \frac{P(i)}{Q(i)}$$

where *P(i)* is the probability distribution of the template frame and *Q(i)* is the probability distribution of any other frame. The result will be a sinusidial wave with minimum and maximum peaks. Ideally, the number of minimum peaks will be the number of wingbeats during the flight. The locations of the mimum peaks will be the frames with probability distributions that match closest to the probability distribution of the template frame. If the template frame is chosen correctly, these are the frames where the bird's shape is closest to the bird's shape in the template frame. The result needs to be smoothed before finding the peaks. MATLAB's function sgolayfilt was used to filter the result, and the function findpeaks was used for finding the minimum peaks.

The results of the KL divergence greatly depends on the template frame chosen. Three template frames tested were at three different postures of the bird: (1) beginning of downstroke (40% correct detection rate), (2) beginning of upstroke (75% correct detection rate), and (3) largest wingspan (95% correct detection rate). The template where the bird had the largest wingspan worked approximtely 95% of the time for the three videos. Therefore, this template frame was chosen for the analysis. The frames that did not match perfectly to the template frame were frames that were very close to the largest wingspan. The algorithm was improved by looking at five frames before and after each minima point of the KL divergence and chosing the frame with the largest wingspan for each wingbeat. After this improvement, I got a 100% correct detection rate for every video.

After computing the KL divergence, I realized that the wing phase could be extracted from this plot (Fig. 3). By inspecting the video and KL divergence, each minimum KL divergence was approximately the midpoint of the downstroke. Similarly, each maxima KL divergence (Fig. 3) was approximately the midpoint of the upstroke. The downstroke and upstroke phases are overlayed in light gray and light blue onto the KL divergence plot in Fig. 3.

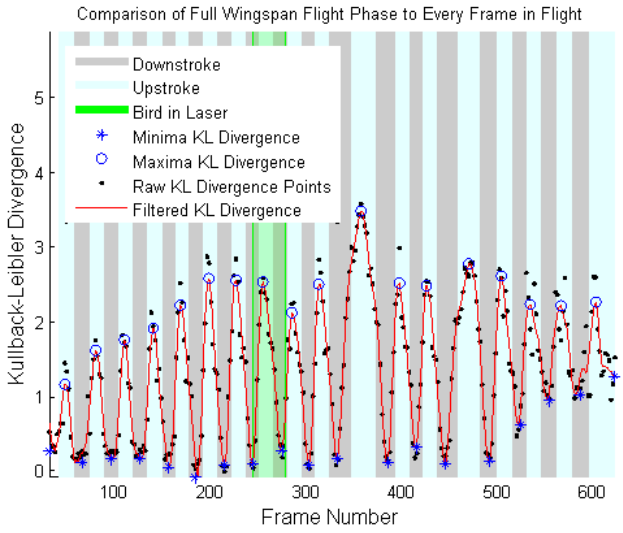

Fig. 3. Kullback-Leibler divergence for flight 1. Each frame represents an instant of the flight. The minima KL divergence are the frames that match closest to the template frame. The green region is the instant when the bird is in the laser sheet during the recording.

After succesfully extracting the flapping frequency and wing phase of each flight, the next important variable that I computed was the forward velocity of the bird. I extracted the forward velocity by doing the following:

1. Measured the real full winspan of the bird.
2. Measured the real length of the takeoff and landing perches.
3. Measured the number of pixels across the takeoff and landing perches in the first frame of the video.
4. Used the numbers obtained in 1, 2, and 3 to relate the length of the perches in pixels to the number of pixels across the bird's full wingspan at the locations of the takeoff and landing perches in the frames.
5. Measured the real distance from the takeoff perch to the landing perch.
6. Related the number of pixels to the distance from the takeoff perch to anywhere between the takeoff perch and the landing perch by using MATLAB's function polyfit and assuming a linear relationship.
7. Used the frames located at the minimum KL divergence to relate the full wingspan length in pixels to the distance between the takeoff perch to anywhere between both perches.
8. Used the central difference formula, in addition to the forward and backwad difference, to obtain the forward velcoity of the bird. A moving average filter was applied to the distances before taking the derivatives.
9. To obtain an estimate of the acceleration of the bird, the speeds were differentiated similiarly to how the distances were differentiated in step 8.

The KL divergence method worked excellent for every flight analyzed. A similar alogorithm to the KL divergence, L1 distance (or taxicab geometry), was also tested. In the L1 distance, the probability distributions are just subtracted. The results for the L1 distance were a little bit less accurate than the results from the KL divergence.

IV. RESULTS AND ANALYSIS

Since very similar results were computed for each of the three flights, results of the first flight are shown.

Fig. 4 shows the flapping frequency as a function of frame number. The frequency plot can be used to pinpoint when the bird starts to bound and for how long. Bounding flight occurs when the flapping frequency decreases significantly. For example, the bird's flapping frequency is relatively constant at around 33 Hz for frames 67 to 303. The bird bounds between frames 300 to 380 and between 440 to 490.

Fig. 5 shows the forward velocity of the bird as a function of frame number. This result shows that the speed is linearly decreasing in speed, which is expected. The speed is also in the range of what is expected.

Fig. 6 shows the forward acceleration of the bird as a function of frame number. The plot shows that the bird is decelerating, which agrees with the velocity plot.

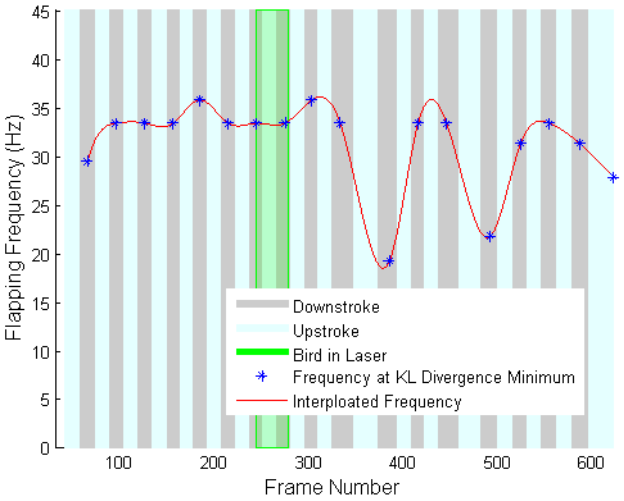

Fig. 4. Flapping frequency as a function of the frame number for flight 1.

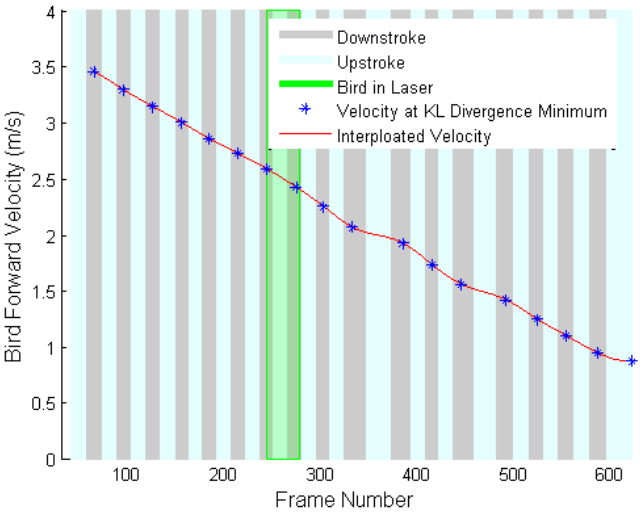

Fig. 5. Bird forward velocity as a function of the frame number for flight 1.

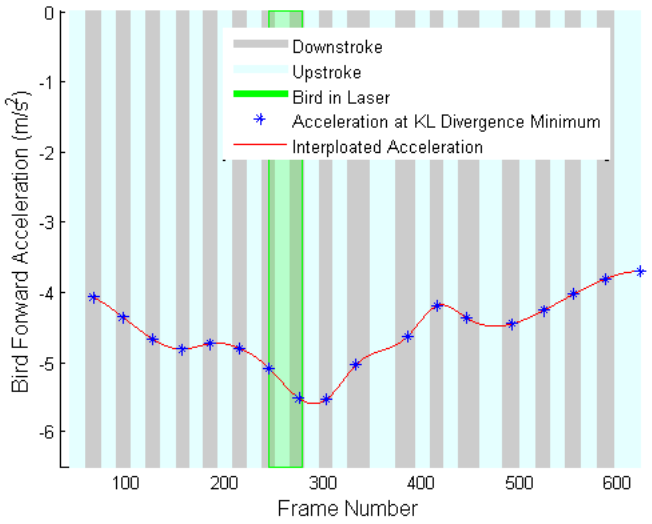

Fig. 6. Forward Acceleration as a function of the frame number for flight 1.

V. CONCLUSION

This study explained an algorithm that was able to successfully (1) extract a moving object (e.g. flapping bird) from a dense background of inhomogeneous illumination, and (2) match a template image (bird with the largest wingspan for a given wingbeat) to every similar frame in the video. The image matching algorithm was able to successfully match the template frame where the bird was at full wingspan to every similar frame. Thus, it had a 100% detection rate for the three flights analyzed. The algorithm is robust changes in position of the bird in the frame and the size of the bird. Since it is able to track the bird even if the bird is out of focus, it is robust to changes in focus of the camera with respect to the bird.

For this study, the camera was positioned at an angle to the flight path of the bird. For very accurate results, the camera needs to be positioned concentric to the flight path of the bird. If the camera is at an angle from the bird's flight path, this needs to be taken into account in the algorithm.

ACKNOWLEDGMENT

I would like to thank my research advisor Prof. David Lentink as well as my EE368/CS232 project mentor Jean-Baptiste Boin. I would also like to thank the course staff for offering a great course on image processing. I would lastly like to thank the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program for supporting me through my graduate studies at Stanford University.

REFERENCES

[1] K.P. Henningsson, G. R. Spedding, and A. Hedenstrom, "Vortex wake and flight kinematics of a swift in cruising flight in a wind tunnel," J. Exp. Biol. 211, 717 – 730, 2008.

[2] T. L. Hedrick, B. W. Toblaske, and A. A. Biewener, "Estimates of circulation and gait change based on a three-dimensional kinematic analysis of flight in cockatiels (nymphicus hollandicus) and rigned turtle-doves (streptopelia risoria)," J. Exp. Biol. 205(10): 1389-1409, 2002.

[3] A. M. Berg and A. A. Biewener, "Wing and body kinematics of takoff and landing flight in the pigeon (Columba livia)," J. Exp. Biol. 213, 1651-1658, 2010.

[4] L.G. Lowe, "Object recogntioon from local scale-invariant features," International Conference on Computer Vision, Corfu, Greece, pp. 1150 – 1157, 1999.

[5] L.G. David, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2, pp. 91 – 110, 2004.