Hovering Hummingbird Automated Segmentation & Feature Tracking

Marc Deetjen
Department of Mechanical Engineering, Stanford University
Stanford, CA
mdeetjen@stanford.edu

***Abstract* – The hovering capabilities of hummingbirds have been studied in literature, and sometimes, image analysis is used to extract important features. Studies using image analysis techniques primarily used multiple cameras whereas here, we wish to conduct image analysis using video from a single camera so that two dimensional kinematics calculations could later be conducted. The benefit of this approach is that a complex, calibrated setup is not necessary and high speed videos of hummingbirds that have already been taken with a single camera can still be analyzed. The process used in this study combines multiple different image processing techniques and the end result is multifaceted and includes segmentation, point tracking, and flapping frequency calculation.**

***Keywords* – Hummingbirds, Image Processing, Image Segmentation, Feature Tracking**

I. Introduction

Hummingbirds are well equipped for hovering flight, and past studies of this behavior have used image analysis to analyze the kinematics of their flapping motion [1], [2]. These studies however, had the benefit of using multiple cameras to pinpoint motion. In this study, video taken with a single camera view is analyzed in order to accomplish multiple tasks including image segmentation, point tracking, wing-beat frequency detection, background movement identification, and angle measurement.

Past studies involving segmentation have used techniques such as normalized cuts [3] and mean shift [4] for image segmentation. These methods however, require some finite image intensity variation across the regions of interest. Because some of the videos of interest are backlit, the entire surface of the bird is mostly uniform and these methods would not succeed for our purposes. For the same reasons, point tracking techniques [5] would be limited.

Based on the limited information available due to potentially poor lighting conditions, along with the increased information available from knowing the subject matter of the video, a customized approach is constructed.

II. Image Processing Algorithm

Processing of the hummingbird analyzed here involved multiple dependent steps and while the processes used could be applied to other birds and situations, the algorithm was optimized for hovering hummingbirds. Further dialogue on future steps to increase the robustness and flexibility of this algorithm is in the Discussion section.

A. Background Movement

To begin, the video was converted to grayscale (figure 1A) after which Otsu's method [6] was applied to calculate an appropriate threshold for converting to a black and white image. A second threshold was also used to generate separate black and white images to help separate the wings, which are slightly transparent, from the body. This threshold was a fraction of the threshold determined through Otsu's method. This assumes that the background is brighter than the bird and works best when the bird is backlit. For certain videos, higher quality is attained if the grayscale image is a color-channel filtered version of the color image. For example, if a front-lit red bird is set against a blue sky, a blue filter would give the largest contrast for the grayscale image.

To find the bird mask in the resulting black and white image, blob detection [7] is used to find the largest area blob which is assumed to be the bird (figure 1B). All other blobs above a threshold area are assumed to be the background (figure 1C). A world origin invariant to camera movement is established by using a sliding window technique [8]. An arbitrary point is defined as the world origin on the first image and subsequent images' backgrounds are compared against the first background to calculate the world origin in each frame. Moving forward, the background is ignored and the two black and white thresholded bird mask images are solely used unless otherwise stated.

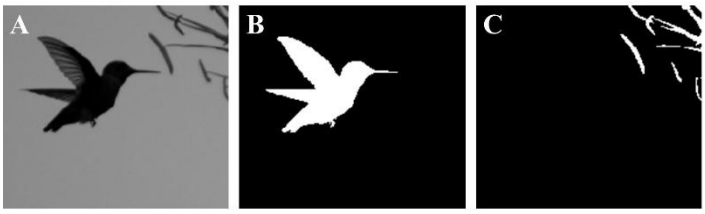

Figure 1: Creation of binary image masks using Otsu's method. (A) Original grayscale image. (B) Bird mask (C) Background mask

B. Tail Detection

To identify the main portion of the body, a couple consecutive frame masks before and after the frame of interest were multiplied together. Assuming the wing movement was much faster than the body movement, this eliminated the wings from the image. For a first approximation of tail position, the vertical center point was found starting at the leftmost portion of the bird which was assumed to be the location of the tail. After a specified number of center points were computed, the approximate tail angle was computed through linear regression (figure 2A).

To increase the accuracy and to fully define the tail, the second black and white thresholded image was used to search for the center of the tail perpendicular to the linear regression. This

second thresholded image was used as it came close to eliminating the wings due to transparency. Center points were computed along this linear regression until a threshold area for the tail was reached at which point the region of the tail was fully defined. A second linear regression was fit to these new center points to accurately define the tail angle (figure 2B).

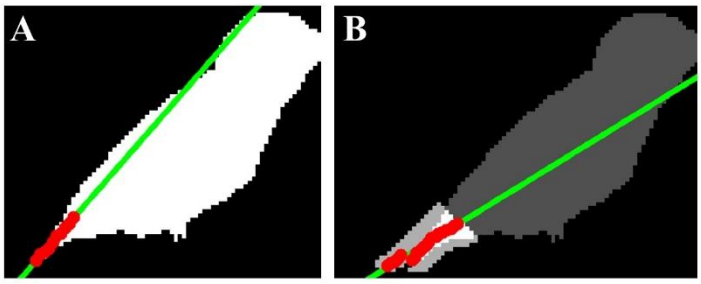

Figure 2: Tail position & angle. (A) First iteration with red dots showing the vertical centerline of the small bird body mask. The green linear regression is used in (B) where red dots show the centerline of the tail along the axis perpendicular to this linear regression. A new green linear regression is used to compute tail angle. Note the light gray tail pixels which are taken from the low Otsu thresholded bird image. The tail area is filled up to a threshold area.

C. Wing Detection

By subtracting the small bird body mask from the bird mask, the approximate wing shape could be found (figure 3A). Most of the time, the two largest blobs were the left and right wings but sometimes, the two wings overlapped and needed to be separated. To determine when this procedure was needed, a histogram of the perimeter of the largest detected blob was created and Otsu's method was used to set a threshold perimeter above which separation was necessary (figure 3B). Perimeter was used because when the two wing blobs touch, their perimeter almost doubles.

To separate the wings, the wing blob was dilated [9] and the overlapping portion of this image with the body image was fit to a linear regression (figure 3C). This line was moved away from the body until there was a gap above a threshold length separating this line into two portions on the left and right wing. The center of the gap was determined and the center of the gap a little further away from the body was also found. These two gap-center points were used to draw a line that separates the wing blob into two separate wings (figure 3D).

After each wing was found, the centroid and tip position were computed. The tip position was found by finding the furthest pixel away from the approximate body centroid. This data was then fit to circles which identified the center of rotation of the wings (figure 3E). This point was used to determine the angle of the wings.

D. Beak Detection

From the small body mask, a first approximation of the beak region was found assuming the beak was at the right side of the image. The horizontal position of the end of the beak was determined as the last column having above a threshold thickness. This beak was then dilated and multiplied by the original bird mask (figure 4A).

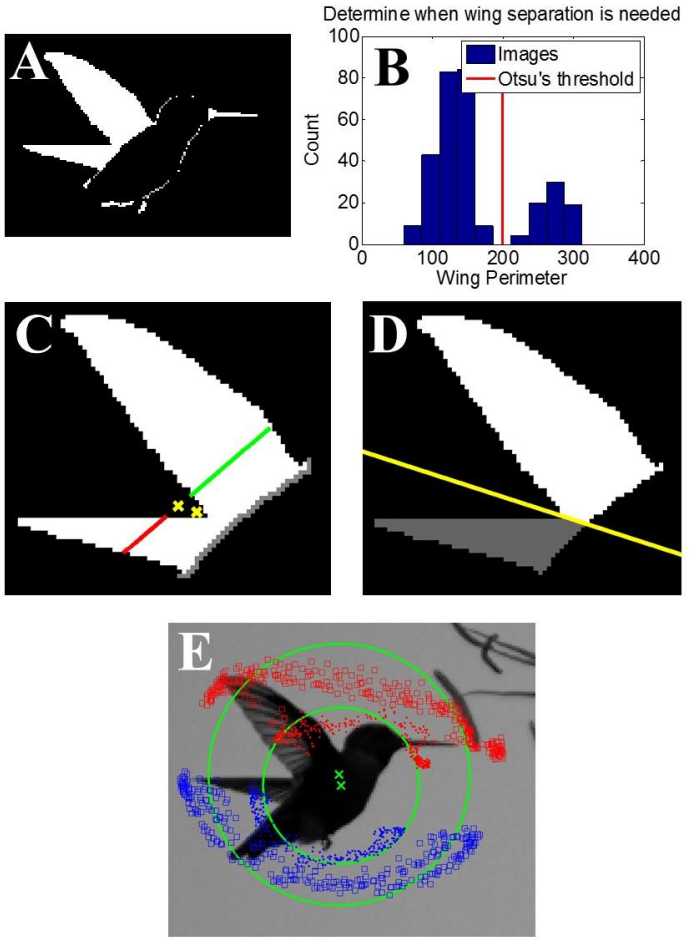

Figure 3: Wing detection, separation, and angle computation. (A) The small bird body mask subtracted from the bird mask to find the wings through blob detection. (B) A histogram of the perimeter of the largest wing blob for all video frames. Otsu's method was used to find a threshold above which the wing blob needed to be separated into two separate wings. (C) The gray region represents the intersection of the dilated wing blob and the bird body. The yellow crosses are used in (D) to separate the wings. (E) The centroid and tip of the wings fit to circles to find the center of rotation at the green cross. The red data correspond to the right wing, the blue data to the left wing, the dots to the centroids, and the squares to the wing tips.

For more accurate beak detection, the sub-pixel center of the beak was found by first taking the Laplacian of Gaussian (LoG) [10] of the grayscale image (figure 4B). Starting at the leftmost beak point, the maximum LoG pixel in that column a couple pixels above and below the approximate beak center position was identified. This pixel and the pixel above and beneath it were fit to a quadratic regression and the maximum vertical position was computed. If the approximate beak mask ended as the algorithm progressed to the right, the next column search region was set as the previous center beak position. Once the curvature of the quadratic fell below a threshold value, the beak was deemed to have ended. The angle of the beak was determined as the linear regression of these beak center points.

For each beak center, the region above and below were checked to ensure isolation from the wings and background. If this was not the case, that center point was deleted. A threshold width of the beak was required to compute the beak angle. If this was not

satisfied, the beak angle was computed as a linear interpolation between surrounding beak angles.

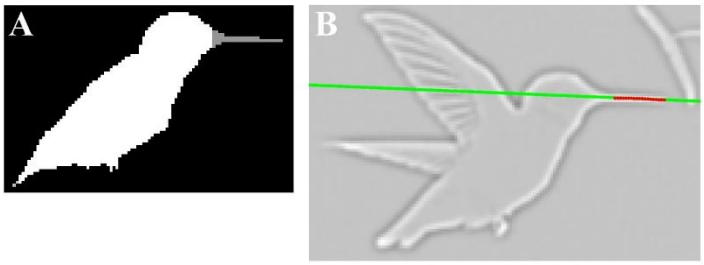

Figure 4: Beak detection. (A) First approximation of beak shown in gray. (B) Laplacian of Gaussian used to find the center of the beak displayed as red dots. A linear regression was fit to these dots to compute the beak angle.

E. Body Center Calculation

The center of the body was calculated using data from the top of the head for a more robust and less noisy signal. The averaged vector from the centroid of the body to the top of the head was computed so that the centroid of the body could be computed using time resolved head data rather than noisy time resolved body centroid data.

In order to compute the position of the top of the head, a Canny edge detector [11] was first applied to the grayscale image (figure 5A). This edge mask was multiplied by the dilated bird body mask to get all the edges on the bird body. From this, the edge with the maximum vertical position was identified along with edges within a threshold vertical distance away as displayed in green (figure 5B). All adjacent pixels with identical vertical positions were eliminated for consistent regression results. Additionally, if the wing was determined to be near to a pixel, it was eliminated. The minimum vertical position on either side of the top of the head was then forced to be identical by eliminating any pixels that made the regression lopsided. Finally, a quadratic regression was computed in order to find the sub-pixel position of the top of the head. This value was interpolated using surrounding data if there were not sufficient data points left for the regression.

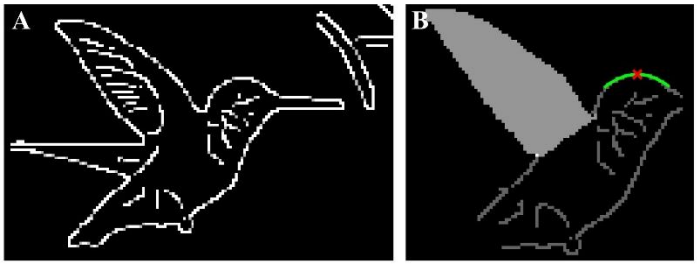

Figure 5: Head center calculation. (A) Edge detection of the grayscale image using a Canny edge detector. (B) Intersection of the Edge detector and the dilated body image as well as the wing displayed in gray. The green line is a quadratic regression fit to the top of the head with the red cross labeling the top point.

III. RESULTS

To visualize the results of this analysis, the original grayscale images (figure 6A) were overlaid with the color-coded segmented regions, angles, center-lines, points of interest that were computed using the algorithm described (figure 6B). Based purely on visual feedback, it can be seen that the algorithm performs quite well and it never completely mislabels a body part in the 300 frame sequence that was analyzed. The primary errors that are observed involve interactions with the wing and the bird body. This can be seen in frame 2 of figure 6B as part of the bird body is identified as the left wing.

Some sample data taken from this video was also plotted in figure 7. The tail angle (figure 7A), body center (figure 7B), and beak angle (figure 7D) raw data are all fit with zero phase-shift 4th order Butterworth filters and it can be seen that the data is relatively smooth and periodic. The same is true for the left and right wing angles (figure 7B) which were fit with sinusoids. While one might observe that the body center data and the beak angle data are noisy, this can be explained by the low magnitude variation and fact that noise is amplified in the calculation of angles. While the accuracy of the data has not been quantified, the visual cues from figure 6B and the consistency of the data indicate that this method is robust for the first video tested.

Another similar, but new video was also tested (figure 6C, 6D) using the automated algorithm. This bird had more front lighting and because of this, did not perform as well. While filtering the colored image with a blue filter improved the performance, there were still regions of the bird that were not well resolved.

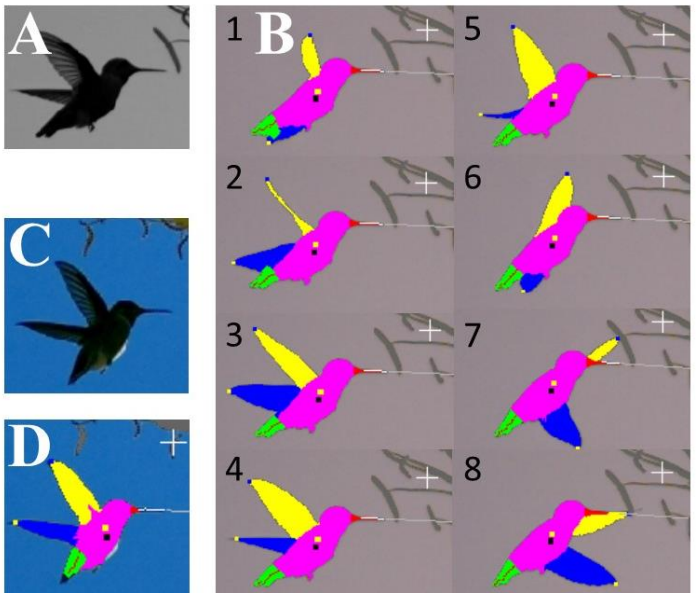

Figure 6: Segmented image. (A) Original grayscale image corresponding to the segmented frames in (B) where the body is shown in purple, the body centroid in black, the beak in red, the beak center in white, the beak angle in gray, the wings, wing tips, and wing center in yellow and blue, the tail in green, the tail center in red, and the offset world origin as the white cross. The black number labels show the progression of frames where every other frame is displayed. (C) Color frame from a different video. (D) Segmented image from this new video.

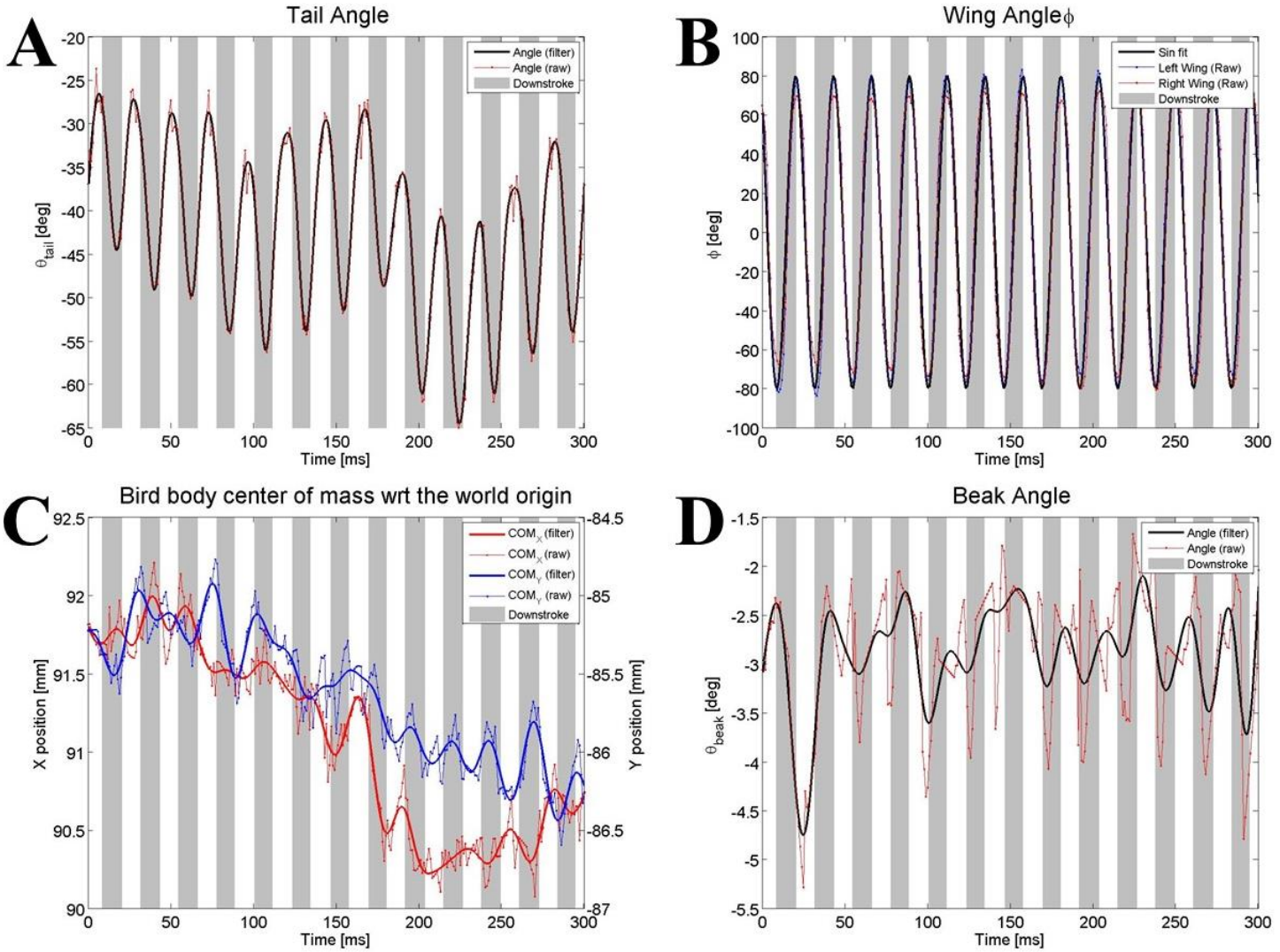

Figure 7: Kinematics data. For all data, the dots are the raw data and the solid lines are the corresponding fitted data. Downstrokes are shaded gray. (A) Tail angle. (B) Wing angle for left and right wings shifted to go from -80 to 80 degrees to match the angle Anna's hummingbird wings travel from an overhead view. (C) Bird body centroid with respect to the world origin. (D) Beak angle.

IV. Discussion

For the primary video analyzed, the results were quite good as verified by visually observing the segmented bird images and by plotting data of key features. Beyond this specific video, while the algorithms still functioned and identified many parts of the bird correctly, there were more errors. Additionally, there are other types of bird videos that the software is currently incapable of tracking.

This software primarily relies on obtaining a noise-free and accurate black and white image of the bird. For a backlit bird, Otsu's method works well to create this binary image, but a more robust bird recognition algorithm would need to be used if the bird was front lit. This could also be extended to detecting the bird given a complex background rather than the current blank sky. While these steps have not been attempted, they would be easy amendments to the beginning steps of this software that would not disturb the rest of the processes.

Additionally, while this software was specifically designed to track hovering hummingbird, the concepts used could be applied to a vast array of bird tracking videos. The major issue to resolve would involve the speed of the bird. One way to approach this problem would be to first find the average position of the entire bird mask as a function of time and move the bird masks to overlap each other when comparing consecutive image to obtain the small bird image without the wings. This method could be challenging if the speed of the bird's wings is slow compared with the movement of other body parts. To solve this issue, an iterative process could be used where after detecting the body position, the movement in each frame could be recomputed and the process restarted. This method would again allow for use of this software as only small changes at the beginning steps would be necessary.

This software was designed to track features on hovering hummingbirds and it accomplished this task well. The data obtained could be used for a two dimensional kinematics analysis and future modifications to improve its robustness & broaden its scope could be easily amended onto the current software.

References

[1] D. L. Altshuler and a. et, "Wingbeat kinematics and motor control of yaw turns in Anna's hummingbirds," *Journal of experimental biology*, pp. 4070-4084, 2012.

[2] B. W. Tobalske and a. et, "Three-dimensional kinematics of hummingbird flight," *Journal of experimental biology*, pp. 2368-2382, 2007.

[3] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.

[4] D. Comaniciu, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, 2002.

[5] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005.

[6] N. Otsu, "Threshold Selectrion Method from Gray-Level Histograms," *IEEE Transactions on systems man and cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[7] J. Sklansky, "Recognition of Convex Blobs," *Pattern Recognition*, vol. 2, no. 1, pp. 3-&, 1970.

[8] E. Ahmed, A. Clark and G. Mohay, "A Novel Sliding Window Based Change Detection Algorithm," in *IFIP International Conference on Network and Parallel Computing*, Shanghai, 2008.

[9] R. Haralick, S. Sternberg and X. Zhuang, "Image-analysis using Mathematical Morphology," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 9, no. 4, pp. 532-550, 1987.

[10] A. Huertas and G. Medioni, "Detection of Intensity changes with subpixel accuracy using Laplacian Gaussian Masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 5, pp. 651-664, 1986.

[11] J. Canny, "A Computational Approach to Edge-detection," *IEEE Transactions on Pattern Analaysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.