Title

Automated three-dimensional body orientation reconstruction and motion tracking with two views during avoidance maneuver of bumblebee

Author

Zhang, Bowen

Publication Date

2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Automated Three-Dimensional Body Orientation Reconstruction And Motion Tracking With Two Views During Avoidance Maneuver of Bumblebee

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Bowen Zhang

Committee in charge:

Professor Nicholas Gravish, Chair
Professor Mauricio de Oliveira
Professor John T. Hwang

2019

The thesis of Bowen Zhang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Prof. Nicholas Gravish for his support during my master research. I'm grateful for the opportunity to work in his lab. His constructive advice and valuable feedback inspired me not only in the research, but also in scientific thinking.

I also would like to thank Dr. Ben Ochoa for his generous help and inspirational ideas in computer vision which arouse my interest and help me overcome obstacles during the research .

I am grateful to Prof. Mauricio De Oliveira and Prof. John Hwang for serving on my master thesis committee and offering their insightful suggestions. I sincerely appreciate their time and efforts.

Special thanks go to all members of the Gravish Lab. I shall always remember all the stimulating discussions and joyful moments we had in the lab.

Last but not least, I would like to thank my family and my friends for the endless support and encouragement throughout my graduate study.

VITA

2017	B. Sc., Mechanical engineering, Dalian University of Technology, Dalian, China
2019	M. Sc., Engineering Sciences (Mechanical Engineering), University of California San Diego

ABSTRACT OF THE THESIS

Automated Three-Dimensional Body Orientation Reconstruction And Motion Tracking With Two Views During Avoidance Maneuver of Bumblebee

by

Bowen Zhang

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2019

Professor Nicholas Gravish, Chair

The bumblebee has excellent performance for flying through complex natural environments and avoiding obstacles. So, it is of interest for engineering and biology to analyze how its flight behaviors are modified during avoidance maneuvers. Thus, we developed a behavior analysis pipeline to auto-track bumblebees and 3D reconstruct its flight motion only in two cameras. The cameras used in this study are calibrated and their poses are linearly and non-linearly estimated under M-estimator Sample and Consensus (MSAC) outliers rejection. The 3D reconstruction is realized by triangulation with root mean square error (r.m.s error) around 1mm. The tracking algorithm is constructed based on

different brightness values of the wing and body. Under image arithmetic and Morphological transformation (like erosion, dilation,etc.), wings and body can be separated from each other. For body orientation estimation, unlikely the traditional way to reconstruct 3D body from 2D images, the bumblebee's body is simplified as a 3D ellipsoid and projected to the image planes. With a defined error function, the projection of ellipsoid will eventually be fitted into the body contours of bumblebee extracted from videos by adjusting the orientation of the ellipsoid in 3D. In this case, only pitch and yaw are considered. The average root mean square error is 3.58 ± 1.21 deg in pitch and 2.84 ± 0.90 deg in yaw across five analyzed videos. Thus, we show that two views are sufficient for this study with a simplified algorithm, comparing to three or more views. The pipeline we developed could accelerate the process of unveiling the flight strategies and biomechanics of insects flight through complex, structured aerial environments.

Chapter 1

Introduction

Flying insects have an extraordinary ability to navigate through the complex natural environments and avoid collisions [Norberg, 1990]. During avoidance maneuver, their flight behaviors are adjusted rapidly according to their inside sensory system, dynamic mechanisms and control mechanisms ([Lin et al., 2014], [Dudley, 2006] and [Sun, 2014]). Therefore, digitalizing and visualizing flight insects' behaviors are essential to get a deep understanding of their maneuver strategies and how those strategies are related to their internal control system. Previously, many studies have contributed to 3D reconstruct flight insects body orientation and wing beats frequency.

1.1 Previous study for 3D body construction

The most common algorithm used for tracking and motion reconstruction is DLTdv5 developed by [Hedrick, 2008], as shown in Figure 1.1. They tracked markers placed on the insect's body to estimate body position. Thus, additional markers needed to be attached to the insect body, and often these markers have to be manually identified and tracked on each frame for 3D tracking results. Besides, for camera calibration, outliers are also rejected manually based on residual error each time in linear estimation. This method requires significant human input and which decreases efficiency and potentially increases error. Even this algorithm can be implemented for three views, it still uses epipolar constraint for every two views instead of the trifocal tensor which could cause underdetermined results in the third view.

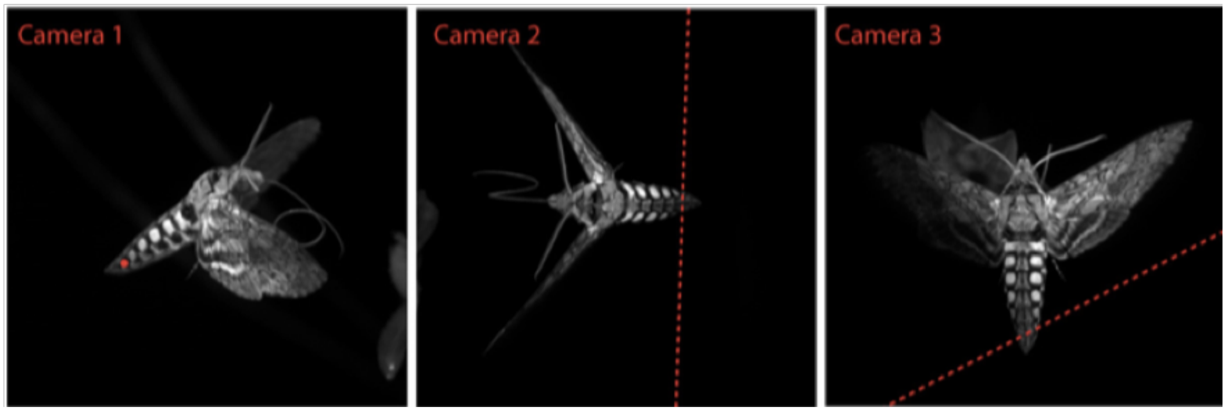

Figure 1.1: DLTdv5 method from [Hedrick, 2008]

Another algorithm was developed for 3D reconstruction of moving animals to estimate its orientation. Hull reconstruction motion tracking (HRMT) was developed by [Ristroph et al., 2009]. By back projecting drosophila aligned silhouettes into 3D space, the visual hull is reconstructed in every frame which reflects body orientation and position, as shown in Figure 1.2. But this method involves complex body construction from 2D to 3D.

Instead of 3D reconstructing dynamic insect's body model in every frame, the 3D physic models of wings and the body are constructed first. Then, those models are projected to images to overlay filmed wings and body by changing the position and attitude of the models. Like [Fontaine et al., 2009], drosophila's geometric model is built via unit quaternions shown in Figure 1.3 and the model construction is too complex. In [Liu and Sun, 2008] and [Fry, 2005] study, as shown in Figure 1.4, they simplified the insect body as a 3D vector which is not accurate. Additionally, during these tracking methods, the real body in the image may be covered by other parts of the body in some image frames

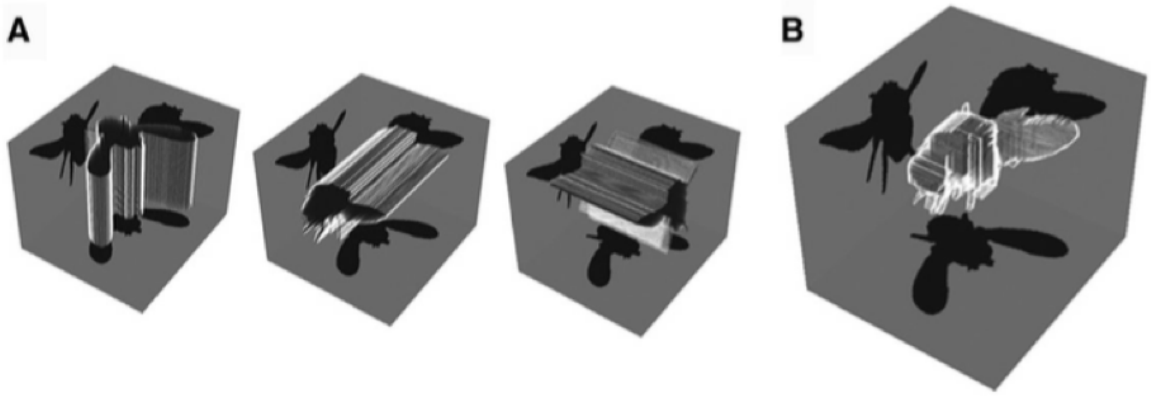

Figure 1.2: Hull reconstruction motion tracking developed by [Ristroph et al., 2009]

which becomes hard to define how well the fitting routine is.

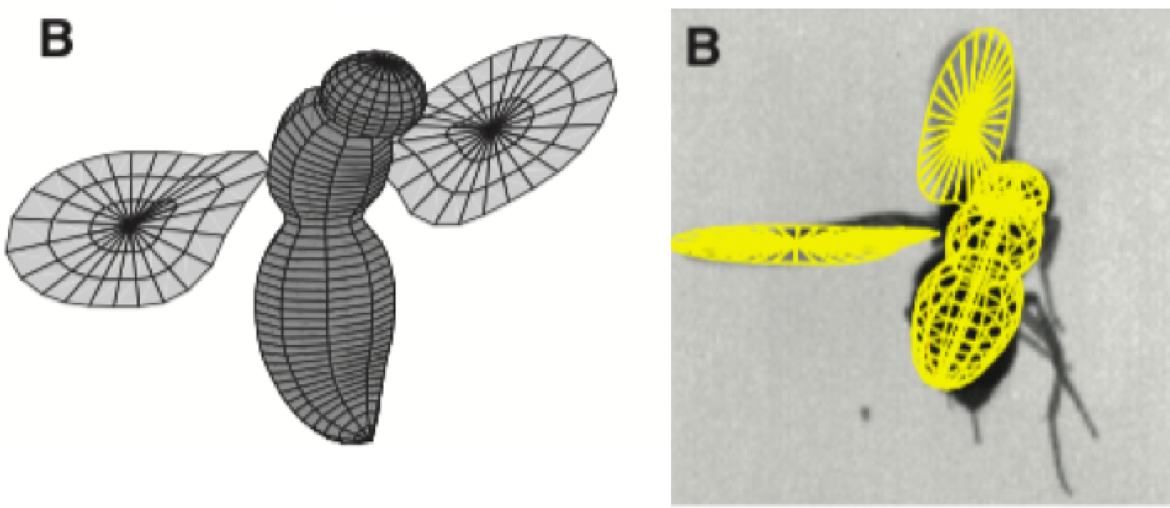

Figure 1.3: Drosophila's geometric model reconstruction via unit quaternions from [Fontaine et al., 2009]

Our approach is motivated by [Liu and Sun, 2008] and [Fontaine et al., 2009]. The bumblebee's body is simplified and constructed first as an ellipsoid. We then project the ellipsoid to images and attempt to minimize the error function we defined. By adjusting 3D ellipsoid's orientation, we fit the projection into the body's silhouette in 2D. In this case, the 3D body's orientation would be measured through the projected ellipsoid orientation.

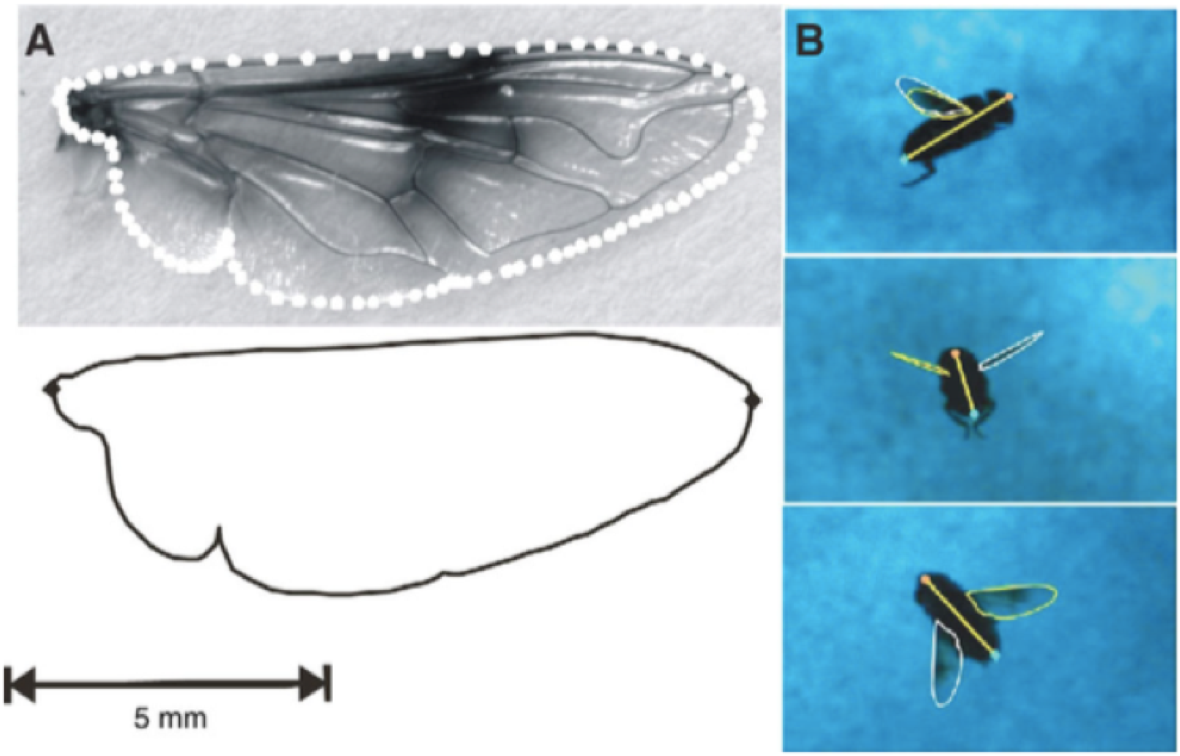

Figure 1.4: Estimate body, wing orientation by fitting physical models into 2D image from [Liu and Sun, 2008] and [Fry, 2005]

We construct this routine for only pitch and yaw for body orientation estimation. In the previous methods discussed above, three cameras are mainly used for 3D reconstruction. In the approach we present here, as few as two views are enough to reconstruct body pitch, yaw and position estimation in three dimensions.

In the following chapter, more details in methods will be discussed about non-linear camera pose estimation, auto outliers rejections, body orientation estimation, and error analysis.

1.2 Background

An image is a rectangular grid of pixels and each pixel usually consists of 1 24 bits to represent the light intensity and color of the image at that pixel. For color images, each pixel usually has 3 channels (R, G, B) with 8 bits each to present 256 kinds of color from 0 (no light) to 255 (full level). If one of the three channels is extracted to form an image, then the color image will turn into a grayscale image with an 8 bits single channel. Comparing to the red and blue channels, the green channel has the best quality. An image with only 1 bit for pixel depth is called a binary image which presents either black(0) or white(1) light intensity.

The morphological transformation is widely used for image processing to modify features on binary images with principal logic operations (AND, OR, NOT, etc). Structuring element (kernel) in this method is a rectangular array to probe an image under study for properties of interest. There are two basic morphological methods: Dilation and Erosion. For dilation, it is used to expand white areas and push out black pixesl among white areas. As the kernel slides through the image, the pixel value is set to 1 as long as there is a pixel value under the kernel is 1 and the pixel value is set to be 0 only when all pixel values under the kernel are 0. For erosion, it has an opposite effect compared to dilation which shrinks the white areas. Because the pixel value is 1 only when all pixel values under the kernel are 1. Typically, either one of these two methods is implemented at first to remove the noise in images. Then, the rest of these two methods is implemented as follows to repair the object's shape without involving additional noise in images. Erosion followed by dilation is called opening, as shown in Figure 1.6. Dilation followed by erosion is called closing, as shown in Figure 1.5.

In the image coordinate system, the image origin is at the top left corner Figure 1.7. The x-axis is pointing rightwards and the y-axis is pointing downwards. However, the camera coordinate system is located at the camera center C. Its z-axis is orthogonal to the

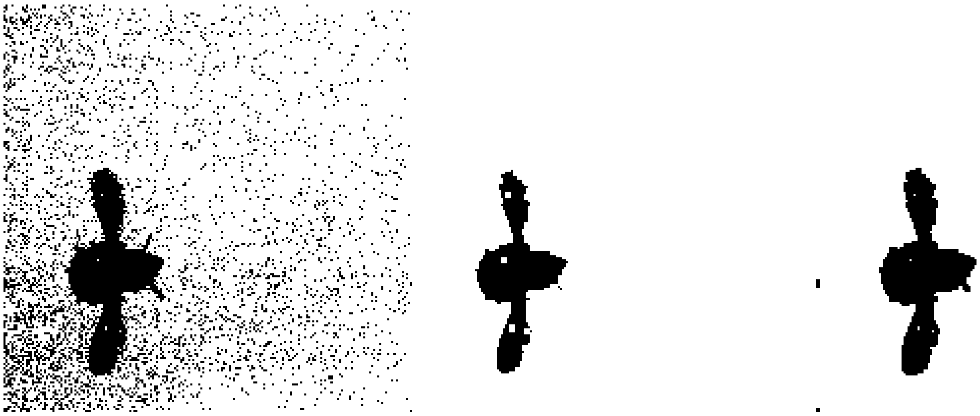

Figure 1.5: Closing: Dilation removes black pixels noise and erosion expands black object's shape

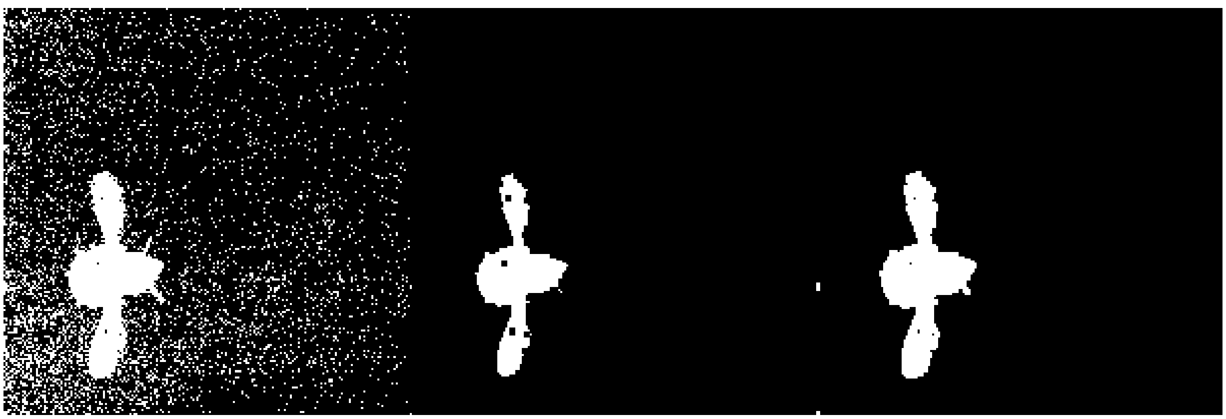

Figure 1.6: Opening: Erosion removes white pixels noise and dilation expands white areas

image plane and the intersection point $(P_x,\ P_y)$ is called the principal point. The distance between the camera center and the image plane is called the focal length. The image plane typically is on the positive z-axis to get an upright image. The following equation is used for covert 3D points in the world frame to 2D points in the image frame.

$$x = K * \begin{bmatrix} R & | & t \end{bmatrix} * X_{world} \tag{1.1}$$

where K is an intrinsic matrix, R and t are rotation and translation between the camera and world coordinate system. All coordinates are represented as homogeneous vectors which in essence attach a "1" to the end of the vector to enable rotation and translation operations through matrix multiplication. To convert homogeneous vectors back to inhomogeneous vectors, the homogeneous vectors are divided by the last element of the vectors, then remove the last element of resulting vectors.

The intrinsic matrix K can be presented as

$$K = \begin{bmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

where f is the focal length but in pixel and $(P_x,\ P_y)$ are the principal points in the image frame.

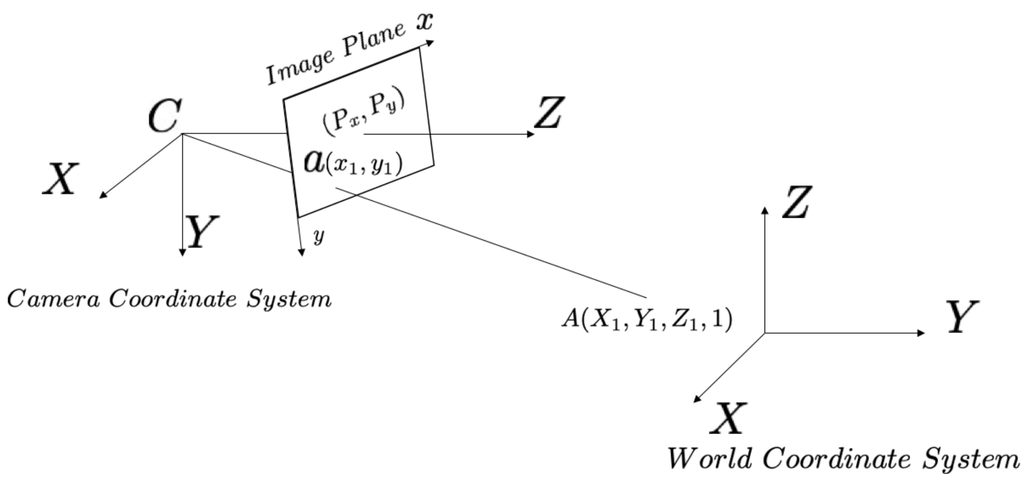

Figure 1.7: Image Plane, Camera coordinate system and World coordinate system.

But when back-projecting 2D image point $(x_1,\ y_1)$ to 3D space, the 3D point $(X_1,\ Y_1,\ Z_1,\ W_1)$ is up to scale and it could be anywhere on the back-project ray. Thus, to get accurate 3D points, a minimum of two back-projected rays/planes or more are needed to generate 3D reconstruction. In the calibrated camera, the intrinsic matrix K is known. So 2D normalized points $\hat{x} = K^{-1}x$ are taken into account instead of 2D points x. For two views, the corresponding normalized points ($\hat{x} \Longleftrightarrow \hat{x}'$) in two images should meet the epipolar constrain Equation 1.3.

$$\hat{x}'^T E \hat{x} = 0 \tag{1.3}$$

where E is an essential matrix. If one of the cameras is the canonical camera with its camera projection matrix is $[I|0]$ and another one is a regular camera with a projection matrix $[R|t]$. The essential matrix would be $[t]_x R$. So normalized point $\hat{x}$ in image 1 generates an epipolar line $l' = E\hat{x}$ on image 2 where the corresponding point in image 2 lies on and same characteristic for epipolar line $l = E^T \hat{x}'$ generated from normalized points in image2 as demonstrated in Figure 1.8. Then back project corresponding normalized points and camera centers to the 3D line. The intersection X of two lines is the reconstructed point in 3D space. This reconstruction method is called triangulation.

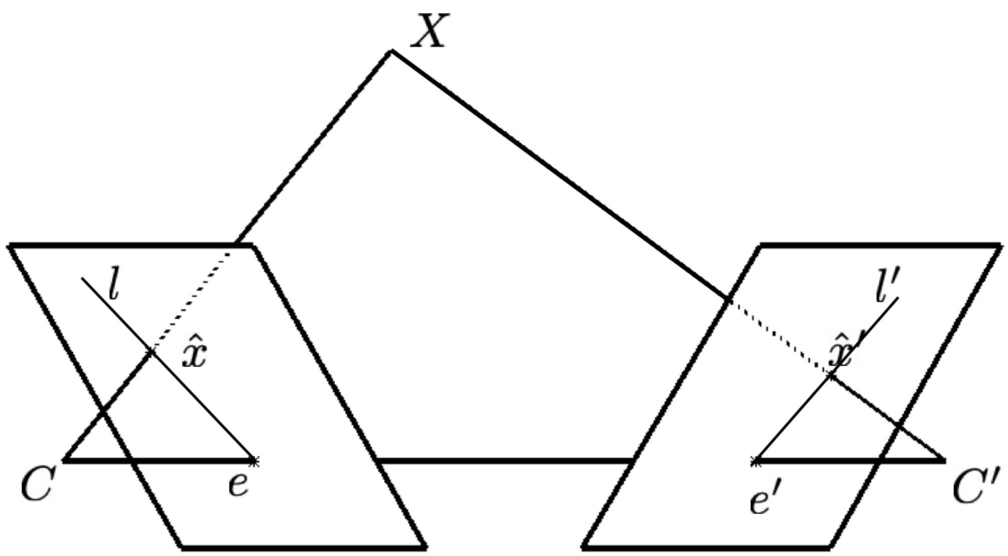

Figure 1.8: Epipolar constraint and 3D reconstruction

Chapter 2

Materials and Methods

A 67cm high obstacle wrapped with checkerboard ($4cm^2$) which improves bumblebee visualized detection [Serres and Ruffier, 2017] stands in the center of the flight chamber ($215cm * 80cm * 81cm$). The enormous chamber leaves enough space for the bumblebee to freely fly around to minimize the optic flow [Baird et al., 2010] and cage size [Stevenson et al., 1995] effect on insect's flight behaviors. Two Phantom VEO410 high-speed cameras (San Diego, CA, USA) placed above the chamber to capture at a resolution of 1028 * 800 with frequency in 5200 Hz Figure 2.1. Bumblebees come from the BioBest company and are released to the flight chamber to fly freely. To auto-trigger high-speed cameras when the bumblebee flies around the obstacle, two Logitech webcams are placed above the obstacle to monitor the same region. If the current frame's pixel values are significantly different from the previous frame's, then it means bumblebee has appeared in the region of interests and Arduino sends a high pulse to cameras and triggers the cameras to capture videos simultaneously.

In the rest of the sections, methods in detail will be discussed and the process of the Pitch-Yaw estimation pipeline is at the following block diagram 2.2.

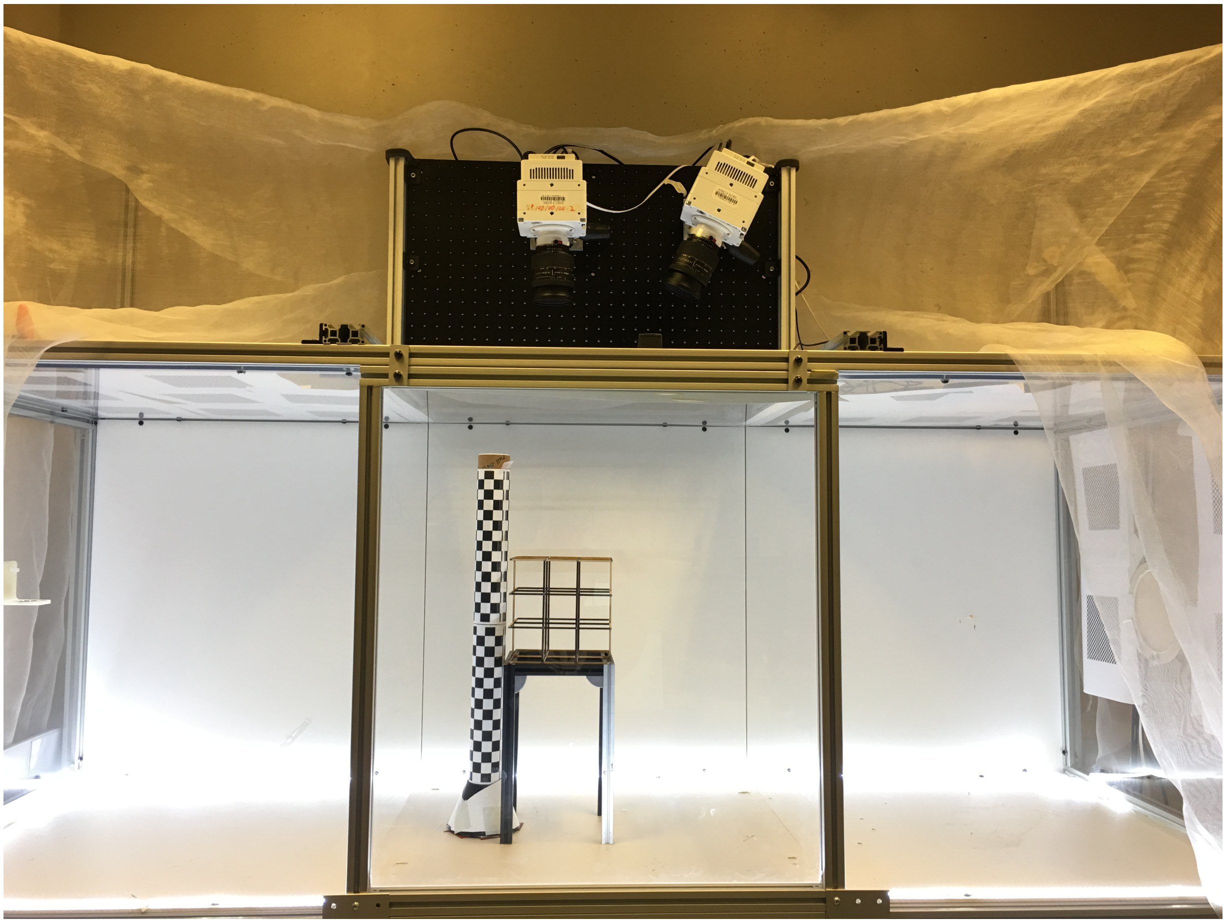

Figure 2.1: Camera1 is right above the obstacle and Camera2 is tilted 45 degrees towards the obstacle to realizing an intersection view.

2.1 Camera calibration and 3D reconstruction

The high-speed camera used in the experiment is Phantom VEO410. Its pixel pitch is 20 $\mu m/pixel$ and the camera center is located at the center of the image. The focal length is set to be 48.33 and 45 mm for Camera1 and Camera2 respectively. Recall the intrinsic matrix K in equation 1.2, in this case, $P_x,\ P_y$ is half of image length and width which is 640 and 400 respectively.

A cube with mortise and tenon joint ($18cm * 18cm * 18cm$) is assembled to collect 3D points in the world frame and corresponding 2D points in the image frame. The global coordinate system is built as shown in Figure 2.3.

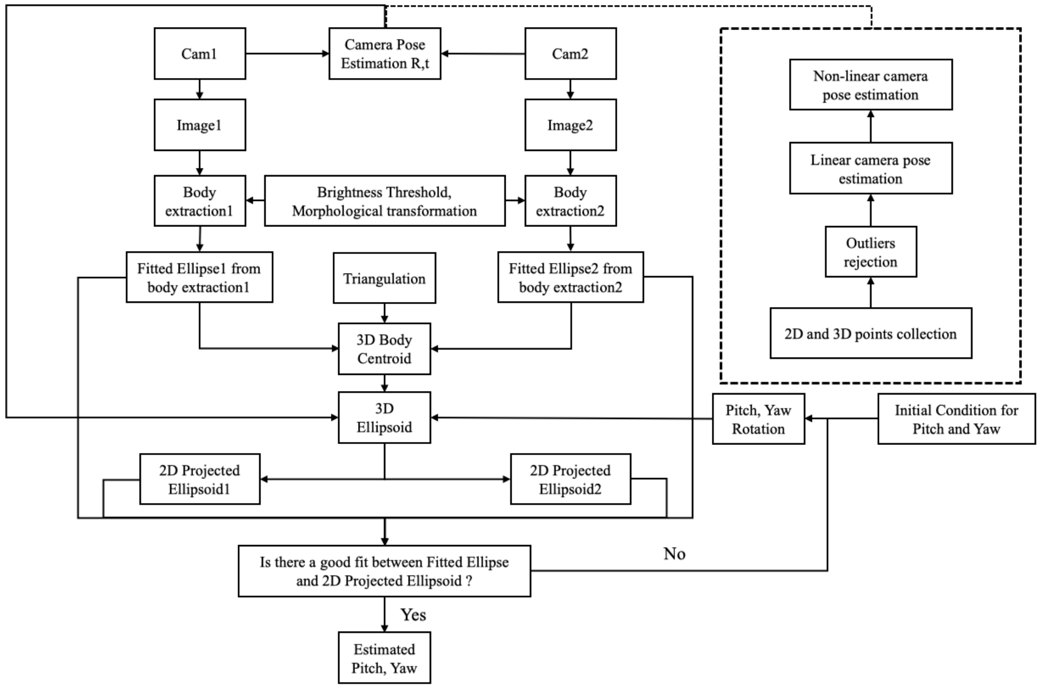

Figure 2.2: Whole process of Pitch-Yaw estimation pipeline

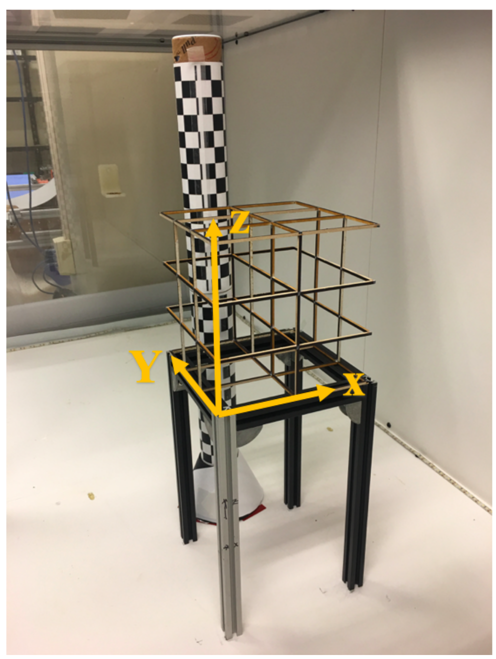

Figure 2.3: The global coordinate system defined. Y and X are along with 1020 aluminum extrusion frame edges. Z axis is perpendicular to the frame plane

With known geometry of the cube, the edge points' coordinate system are computed. And those corresponding points' in the 2D image are gathered by clicking points on image shown in Figure 2.4

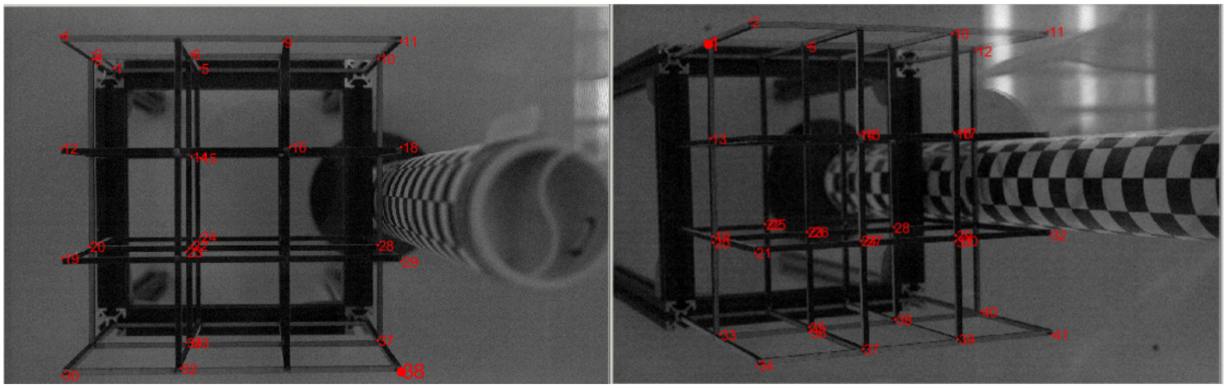

Figure 2.4: The cube with tenon and mortise joint viewed under camera1 (left) and camera 2 (right) respectively. Manually clicked points lied on the edge of cube marked as red to collect 2D points under Camera 1 and Camera 2 respectively.

2.1.1 Outliers rejection

For each point in image frame, its point in normalized coordinate is calculated $\hat{x} = K^{-1}x$. Then the MSAC algorithm [Murray and Road, 1997] is applied to reject the outliers. For each trial, randomly select three points from the 2D dataset and world 3D dataset respectively. 3-point algorithm of Finsterwalder reviewed by [Haralick, 2000] is used to find those three corresponding points in the camera frame. And [Umeyama, 1991] algorithm is applied to estimate camera pose rotation R and translation t from the coordinates of camera frame to world frame. With estimated transformation R and t, project all 3D points in world frame to normalized image frame and computed projection error which is sum square error (s.s.e.) $(\sum_{k=1}^{n}(\hat{x} - \begin{bmatrix} R & | & t \end{bmatrix} X_{world}))^2$. For the projection error of each point in the 2D normalized frame, if this error is less than squared distance threshold t^2, then this point is labeled as an inlier and vice versa. The squared distance threshold is defined as :

$$t^2 = F_m^{-1}(\alpha)\sigma^2 \tag{2.1}$$

where $F_m^{-1}(\alpha)$ is the inverse chi-squared cumulative distribution function with $m = 2$ degree of freedom at the probability $\alpha = 0.95$ and variance σ^2 assumed to be 1. The initial maximum trial for MSAC is set to be infinity. But after each trail, update the value of maximum trial and save the inliers index with minimum projection error up to now.

$$max\_trial = \frac{ln(1-p)}{ln(1-w^s)} \tag{2.2}$$

where p is the probability that at least one of the random samples does not contain any outliers and is chosen to be 0.99. s is the sample size and w is the probability that a data point is an inlier $w = \frac{Total\ number\ of\ inliers}{Total\ number\ of\ data\ points}$. If the current number of iteration is greater than the maximum trial, iteration stops and returns the inliers index.

2.1.2 Pose estimation

After outliers rejection, we implement the Efficient Perspective-n-Point(EPnP) method from [Lepetit et al., 2009] is applied to estimate all inliers' 3D points X_{cam} in the camera frame with known 2D points in the image plane and 3D points in world frame X_{world}. The Umeyama algorithm is then used to linearly estimate normalized camera projection matrix $\hat{p} = \begin{bmatrix} R & | & t \end{bmatrix}$ with corresponding points in the camera frame and world frame obtained from the EPnP method.

The Levenberg-Marquardt(LM) algorithm for this particular case ([Levenberg, 1943] and [Marquardt, 1963]) is developed to non-linearly estimate parameter vectors with global minimal measurement vector error. In this case, measurement vectors are estimated 2D normalized points $(\tilde{\hat{x}}_1^T, \tilde{\hat{x}}_2^T ... \tilde{\hat{x}}_n^T)^T$ and parameter vectors are estimated rotation and translation $\tilde{\hat{p}}^T = (\tilde{R}^T, \tilde{t}^T)^T$.

$$\hat{x} \approx \tilde{\hat{x}} + J(\hat{p} - \tilde{\hat{p}}) \tag{2.3}$$

where $J = \frac{\partial \tilde{\hat{x}}}{\partial (R,t)}$, $\hat{x}$ is the ground truth 2D normalized points and $\hat{p}$ is the ground truth transformation matrix. This equation can be represented as

$$\epsilon = J\sigma \tag{2.4}$$

where $\epsilon = \hat{x} - \tilde{\hat{x}}$ and $\sigma = \hat{p} - \tilde{\hat{p}}$. Thus, the new estimated transformation matrix is $\hat{p} + \sigma$

To avoid singularities and for improved convergence and locally continuous while tracing down the global minimal point, minimal parameterization has been applied which parameterize the camera rotation R with angle-axis representation of rotation matrix $w = \theta\hat{v}$, the transformation can be represented as

$$\begin{bmatrix} e^{[w]_x} & | & t \end{bmatrix} \tag{2.5}$$

where $[w]_x = ln(R)$

Thus, the normalized projection matrix becomes $\hat{p} = [w|t]$ from $[R|t]$, which has 6 degree of freedom. And $J = \frac{\partial \tilde{\hat{x}}}{\partial (w,t)^T}$.

To convert R to w:

$$sin(\theta) = \frac{v^T \hat{v}}{2} \tag{2.6}$$

where $\hat{v} = \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}$

$$cos(\theta) = \frac{Tr(R) - 1}{2} \tag{2.7}$$

$$\theta = tan^{-1}(\frac{sin(\theta)}{cos(\theta)}) \tag{2.8}$$

if θ<0:

$$\theta = \theta + 2\pi \tag{2.9}$$

if $\theta > \pi$:

$$w = w(1 - \frac{2\pi}{\theta}\lceil\frac{\theta - \pi}{2\pi}\rceil) \tag{2.10}$$

To convert w to R:

$$R = cos(\theta)I + sinc(\theta)[w]_x + \frac{1 - cos(\theta)}{\theta^2}ww^T \tag{2.11}$$

The LM algorithm is to locate the global minimal point for equation $\epsilon = J\sigma$ to minimize projection error and estimate transformation matrix. First, construct the normal equation

$$(J^T J)\sigma = J^T \epsilon \tag{2.12}$$

Then, weight least-square problems normal equation

$$(J^T \Sigma_x^{-1} J)\sigma = J^T \Sigma_x^{-1} \epsilon \tag{2.13}$$

Last, augment normal equation

$$(J^T \Sigma_x^{-1} J + \lambda I)\sigma = J^T \Sigma_x^{-1} \epsilon \tag{2.14}$$

The initial λ and projection error is set to be 0.001 and infinity respectively. After one iteration, if it returns a smaller projection error than the previous one, save and update $\hat{p} = \tilde{\hat{p}} + \sigma$ and substitute $\lambda = 0.1\lambda$ to decrease step size and converge closer to the minimal point. Otherwise increase step size by substituting $\lambda = 10\lambda$. And the biggest cost reduction happens after the 1st iteration.

After outliers rejection and non-linear pose estimation, projection results are shown in Figure 2.5

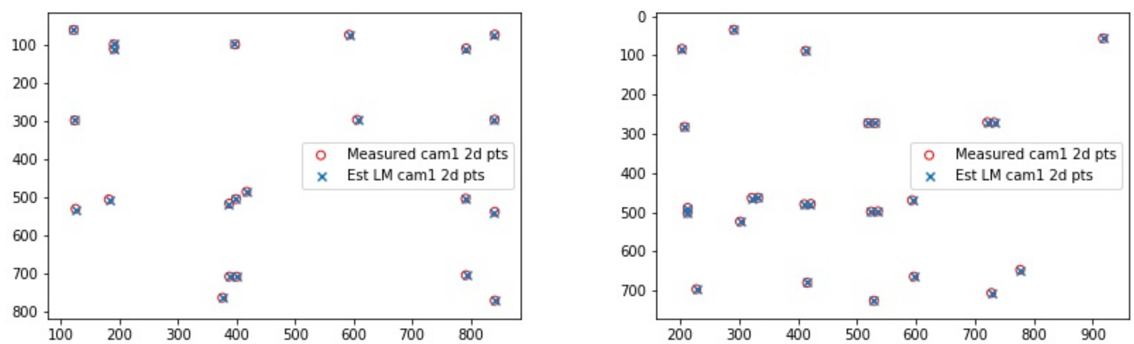

Figure 2.5: Red circles are actual points lied on the edge of the cube in the images and blue markers are their 3D projected points. Root mean square projection errors are 1.26 mm and 1.13 mm among 22 and 25 inliers for camera1 and camera2 respectively

2.1.3 3D reconstruction

To reconstruct 3D points, the corresponding points ($\hat{x} \Longleftrightarrow \hat{x}'$) need to be generated which meet the epipolar constraint. And the essential matrix E also is constructed. In this case, $\hat{p} = [R|t]$ and $\hat{p}' = [R'|t']$. Thus, to compute the essential matrix with form $[t]_x R$, the transformation should be converted as follows: $\hat{p} H_E^{-1} = [I|0]$ and $\hat{p}' H_E^{-1} = [R_{new}|t_{new}]$ And,

$$\hat{p} = [I|0] \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} = [I|0] H_E \tag{2.15}$$

$$H_E^{-1} = \begin{bmatrix} R^T & -R^T t \\ 0^T & 1 \end{bmatrix} \tag{2.16}$$

Therefore,

$$\hat{p}' H_E^{-1} = [R' R^T | t' - R' R^T t] = [R_{new}|t_{new}] \tag{2.17}$$

Essential matrix is

$$E = [t_{new}]_x R_{new} = [t' - R'Rt]_x R'R^T \tag{2.18}$$

The 3D reconstruction from two cameras is to back project two rays which pass through camera center and corresponding 2D normalized points $\hat{x} \leftrightarrow \hat{x}'$ so that these two rays intersect at some point in 3D space. That intersected 3D point is the reconstructed 3D point based on two corresponding 2D normalized points. But due to noise in corresponding points $\hat{x} \leftrightarrow \hat{x}'$, They fail to meet epipolar constraint. Thus, back projected two rays most likely are not intersect with each other in the 3D space Figure 2.6. Based on [Hartley et al., 1996], shown as Figure 2.7, the correspondence normalized points are corrected to be nearest points $(\hat{x}, \hat{x}') \implies (\tilde{\hat{x}}, \tilde{\hat{x}}')$ lie on the epipolar line with minimal sum squared distance error $d(\hat{x}, \tilde{\hat{x}})^2 + d(\hat{x}', \tilde{\hat{x}}')^2$.

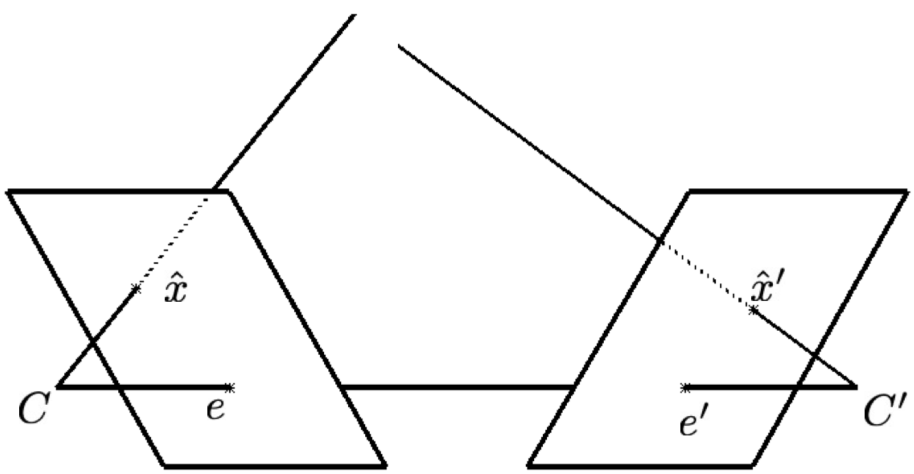

Figure 2.6: Back rays are not likely intersected at 3D space due to noise in 2D normalized points $\hat{x} \leftrightarrow \hat{x}'$

After correcting 2D normalized points, 3D point X can be found by triangulation method. First find an orthogonal line $l'_\perp$ to $l' = (a', b', c')^T$ which pass through $\hat{x} =$

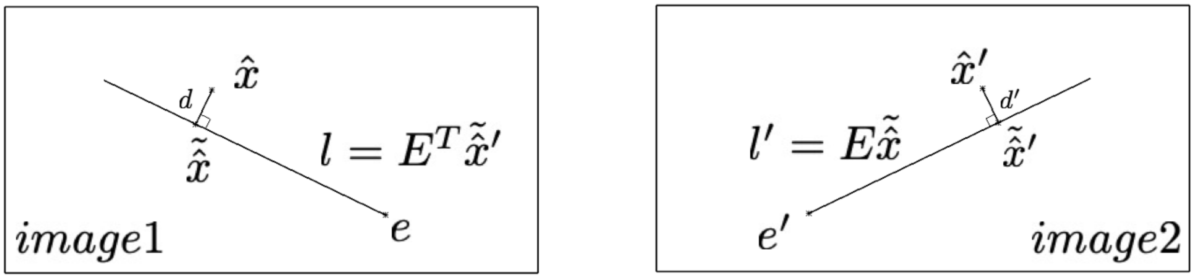

Figure 2.7: Correct $\hat{x}$ and $\hat{x}$ to the nearest points $\tilde{\hat{x}}$ and $\tilde{\hat{x}}'$ which meet the epipolar constraint

$(x', y', w')^T$.

$$l'_{\perp} = (-b'w', a'w', b) \tag{2.19}$$

Then, back project $l'_{\perp}$ to 3D plane π. And the intersection of plane π and camera1 back projected ray is the reconstructed 3D point Figure 2.8.

After 3D reconstructing 8 corresponding points on the calibration cube, the r.m.s error is 0.876 mm.

2.2 Body and wing extraction

To analyze bumblebee motion behaviors during the avoidance maneuver, body and wing have been extracted separately. Due to the different amounts of illumination from the wing, body and background reflect back to the cameras, different brightness thresholds are applied to extract body and wing. Under Morphological transformation, it removes noise and improves resolution. Based on body and wing contours, fitEllipse function from OpenCV is applied to find the best-fitted ellipse to the contours with the least square algorithm.

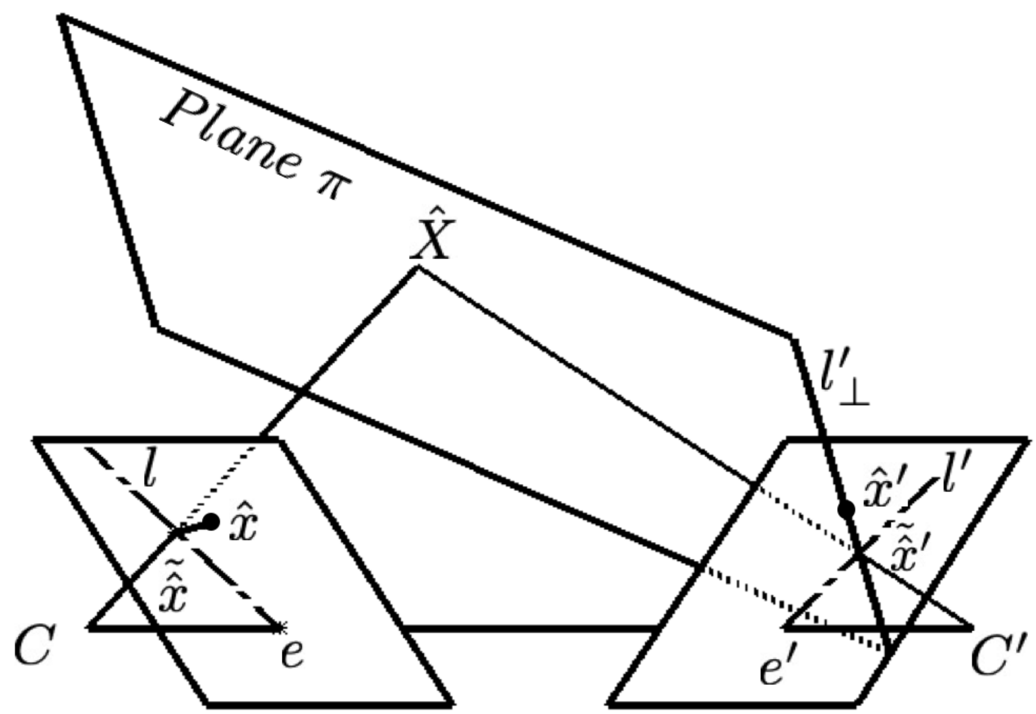

Figure 2.8: 3D point reconstruction from corrected points $\tilde{\hat{x}}$ and $\tilde{\hat{x}}'$ with the intersection of line and plane

2.2.1 Brightness threshold

The videos return from high-speed camera are grayscale videos. In grayscale videos, there is only one channel for each pixel which reflects its brightness. Because wing and body have different brightness values compared to the background. To extract body and wing separately, different brightness threshold is applied to each frame. If a pixel value is greater than the threshold, then it is converted to 1 which is a white dot. Otherwise, it will be converted to 0 which is a black dot. Typically, $brightness_{background} > brightness_{wing} > brightness_{body}$. Therefore, in my analysis, two different thresholds are applied. One is $(15 \sim 18)$ to extract bumblebee body and wing from background Figure 2.9 and another is $(30 \sim 40)$ to extract bumblebee body only. But for the following step to extract wings only, the threshold method for body extraction is implemented in the opposite way Figure 2.10

which means higher pixel values will be converted to 0 and lower ones will be converted to 1. In the following subsection, more details will be discussed to illustrate the reason to convert pixel value in the opposite way.

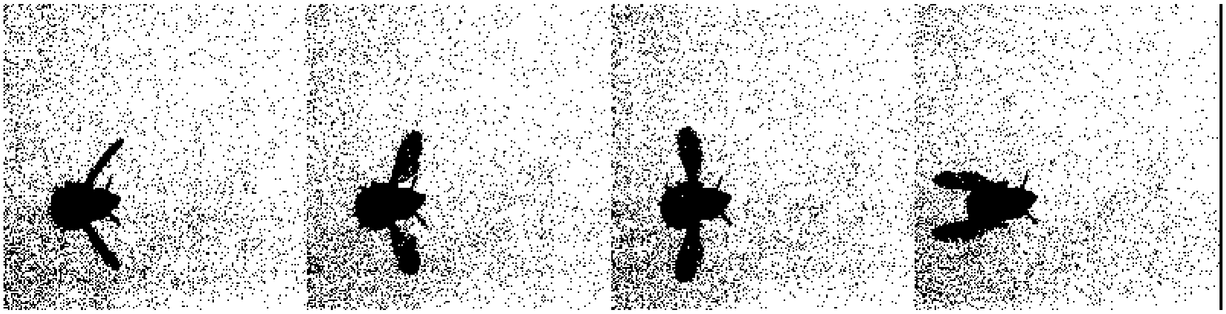

Figure 2.9: Resulting frames after the brightness threshold method ($15 \sim 18$)

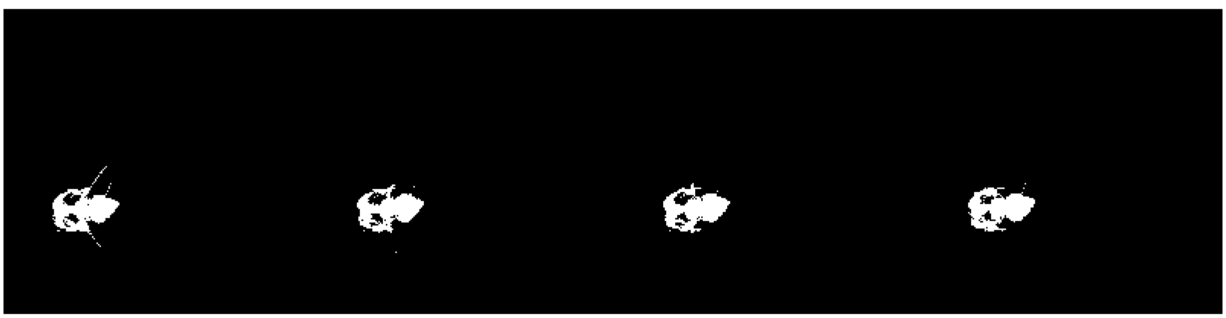

Figure 2.10: Resulting frames after the inverse brightness threshold method ($30 \sim 40$)

2.2.2 Morphological transformation

After thresholding the images, the resulting frames still have noise in the form of small speckled white pixels in additino to the wings or body. To remove noise properly and extract the wings only, the morphological transformation has been applied. From Figure 2.10, to remove black pixels in the background, closing (dilation followed by erosion) from OpenCV has been applied with window size $3*3$. In dilation, if there is at least one pixel under the window is 1, then all pixel values are set to be 1 (white dots) which can push out black pixels among white areas. To prevent shrinking of the object's shape after dilation, then erosion is applied. Different from dilation, if there exists at least one pixel smaller than 1, then all pixels are set to be 0 (black) which only increases existed black areas of the object without involving in black dots noise. The results are shown as the following Figure 2.11.

Figure 2.11: Resulting frames after closing (dilation after erosion) under window size $(3*3)$

To only extract wings, results from Figure 2.10 and Figure 2.11 are add up together so that black body areas in Figure 2.11 will be covered by white body area in Figure 2.10 and will blend into the white background and leave black wings only. But before summing them up, dilation has been applied to results in Figure 2.10 to enlarge white body areas and avoid remained body contour after summing up method shown in Figure 2.12

Figure 2.12: Resulting frames after dilation underwindow size $(3*3)$

After the dilation, resulting frames are shown as Figure 2.13

To avoid piecewise wings after the previous dilation, the opening method is implemented to push out white dots in the wing and keep its structural integrity Figure

Figure 2.13: The add-up method for wing extraction

2.14.

Figure 2.14: The opening method to maintain wings' completeness

In order to auto-track wings and body separately, findContours and fitEllipse are implemented from the OpenCV library. Body and wings contours are generated from Figure 2.10 and Figure 2.14. During drawing contours process, some additional contours could be generated either from image boundary or noise. Therefore, to target the exact contours for body and wings, contours' areas are considered to judge if a contour belongs to either wing or body. For the body contours, the contour area is usually located in range (300,2500). For the wings contours, their areas are usually the top two areas among all contours. With fitEllipse function, a best-fitted ellipse is created to describe contour's shape via the least square algorithm. the results are shown in Figure 2.15. With fitted ellipses, they can provide body, wing orientation and body centroid in the 2D image which is essential for later 3D flight trajectory reconstruction, 3D body orientation reconstruction, and wing beat frequency analysis.

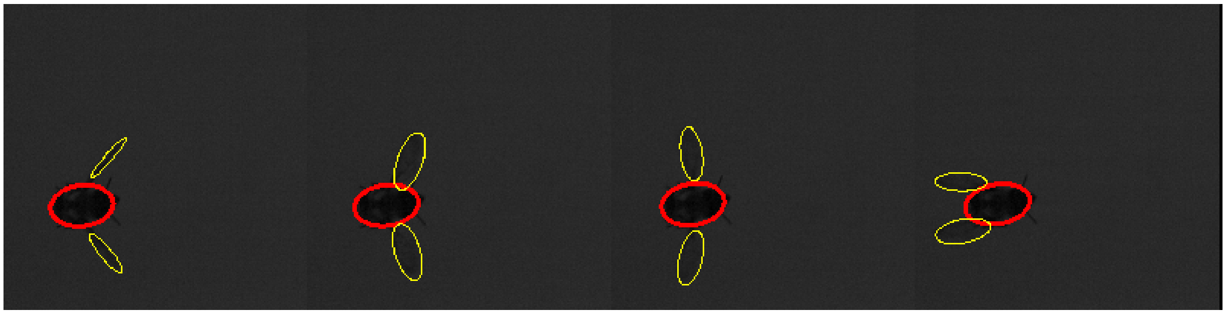

Figure 2.15: Generate an ellipse to best describe the body's and wings' contours

2.3 Body orientation estimation

An ellipsoid is used to describe the simplified bumblebee body shape in 3D. With fitted ellipse obtained from body contour, the ellipse center can be considered as bumblebee centroid. Therefore, with two corresponding centers in 2D, the 3D bumblebee centroid is reconstructed for each frame under triangulation discussed before where 3D ellipsoid's center $X_{center} = (X,\ Y,\ Z)^T$ is located at. Project ellipsoid to 2D image and adjust ellipsoid's orientation in 3D to minimize error function and estimate bumble body orientation in 3D.

2.3.1 3D ellipsoid projection

The local coordinate system for 3D ellipsoid is built by translating global coordinate system 2.3 from its origin to bumblebee's centroid. The initial ellipsoid's principal semi-axes $a,\ b,\ c$ are along with its local coordinate system $x,\ y,\ z$ respectively which stands for bumblebee body width, length, height. Rotation around y-axis is defined as roll, rotation around x-axis is defined as pitch and rotation around z-axis is defined as yaw. In this study, only pitch and yaw are estimated for its body orientation,

To project ellipsoid to the image plane, the surface points of the ellipsoid are presented as follows.

$$
\begin{aligned}
x &= a\sin(\theta)\cos(\phi) + X \\
y &= b\sin(\theta)\sin(\phi) + Y \\
z &= c\cos(\theta) + Z
\end{aligned}
\tag{2.20}
$$

From the known intrinsic matrix K and extrinsic matrix $\hat{P}$ of two cameras, all surface points $X_{surface}$ are projected to the 2D image plane shown in Figure 2.16.

$$
x = K\hat{P}X_{surface} \tag{2.21}
$$

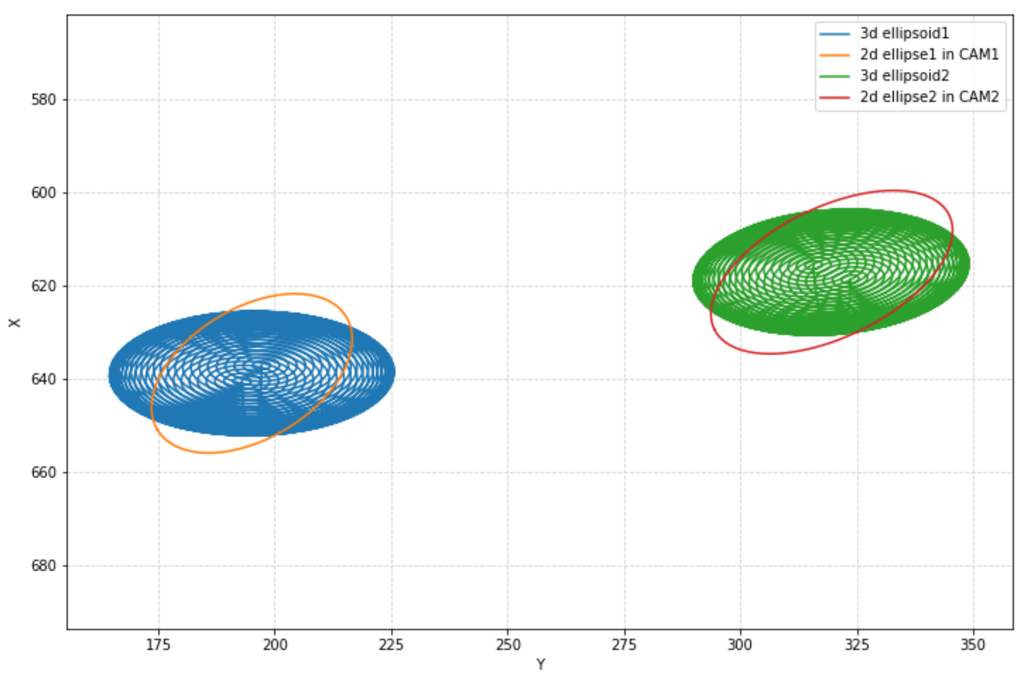

Figure 2.16: The ellipsoid surface points project to the 2D image plane. Blue and green points come from ellipsoid and orange and red ellipses come from the body contours

As Figure 2.16 shown, each projected ellipsoid in 2D can be fitted into an ellipse to best describe its contour. But in order to apply fitEllipse function from OpenCV, all points in float data type should be converted to the int data type. The fitted results are shown in Figure 2.17

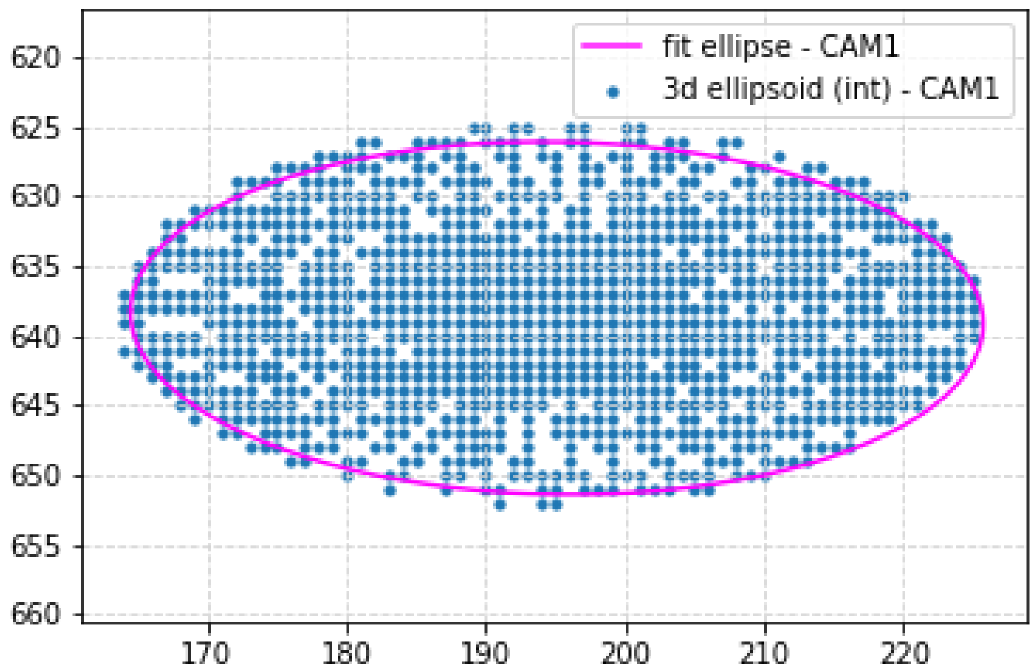

Figure 2.17: Best-fitted ellipse to ellipsoid in 2D

With this fitted ellipse, it can return the orientation, major, minor axis of the projected ellipsoid in 2D which is essential for the following error function defined.

2.3.2 Error function defined

To demonstrate how well the ellipsoid in 2D ($ELPS_2$) fits into body contour ($ELPS_1$) extracted from videos. An error function has been defined shown in Figure 2.18.

There are two parts of the error function. One is distance error (Δd) related to pitch angle. The distance error is the distance difference between the semi-major axis of $ELPS_1$ and the projection of semi-major axis of $ELPS_2$ along image x-axis. Another one

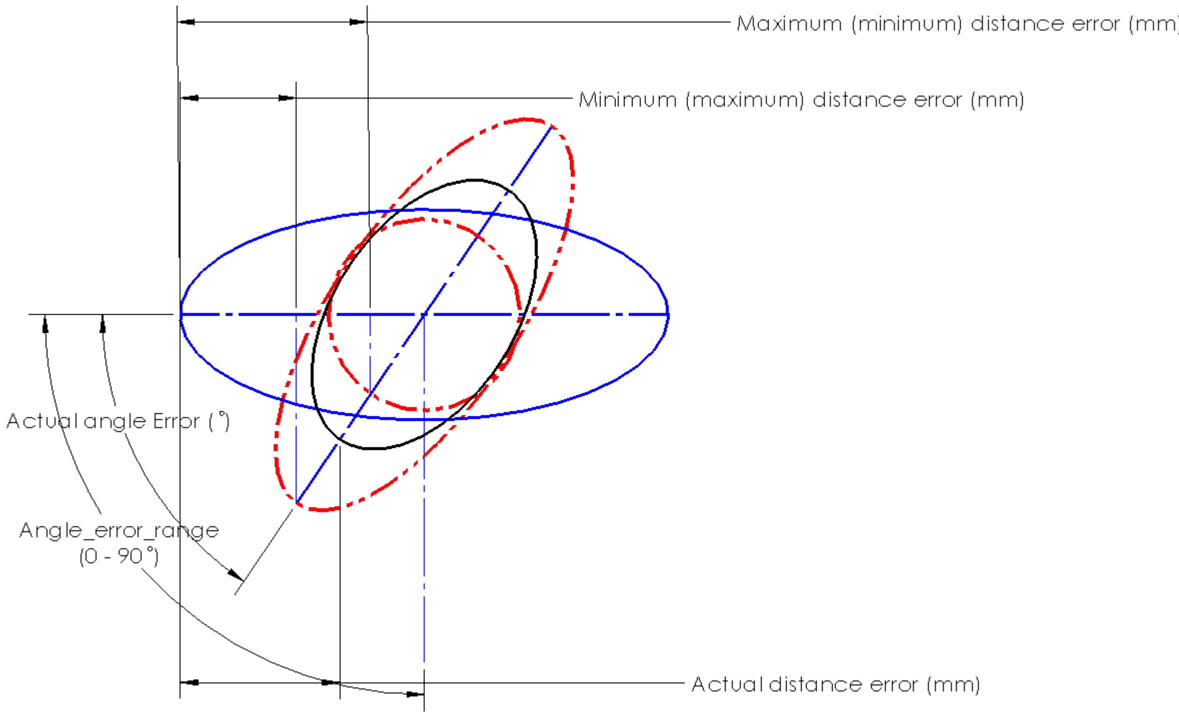

Figure 2.18: Error Defined, the blue ellipse ($ELPS_1$) comes from the body contour, the black ellipse ($ELPS_2$) comes from the projected 3D ellipsoid and two red dash ellipses have maximum/minimum distance error.

is angle error (α) related to yaw angle. Angle error ($0 - 90°$) is the angle between two major axes of $ELPS_1$ and $ELPS_2$. To ensure unit consistency for two kinds of error (angle and distance), error percentage is involved for each kind of error to fix this problem which has range from 0 1, where 0 means 0 offset from ground truth and 1 means the greatest deviation. For the angle error, in the worst case, the major axis of $ELPS_1$ is perpendicular to $ELPS_2$ major axis where maximum angle error is 90°. Thus, the percentage of angle error is

$$PCT_{angle} = \frac{\alpha}{90°} \tag{2.22}$$

For distance error, the maximum deviation occurs at either $ELPS_2$ is fully extended and its length of the major axis is equal to the body length in 2D or $ELPS_2$ becomes a circle eventually and its length of the major axis is equal to the body width in 2D. The maximum distance error is defined as follows.

$$MAX_{distance} = |(a1 - a2 * \cos(\alpha))| or |(a1 - b2 * \cos(\alpha))| \tag{2.23}$$

where a1 is the semi-major axis of $ELPS_1$ and $a2, b2$ are the semi-major and semi-minor axis of $ELPS_2$ respectively. And the percentage of distance error is

$$PCT_{distance} = \frac{\Delta d}{MAX_{distance}} \tag{2.24}$$

Thus, the final error function is

$$f(\Delta d, \alpha) = PCT_{distance} + PCT_{angle} \tag{2.25}$$

2.3.3 Optimization function defined

The optimization in this study is the Nelder–Mead method from python scipy.optimize package. Pitch and yaw are as inputs to minimize the error function.

$$\tilde{f}(yaw, pitch) = PCT_{distance} + PCT_{angle} \tag{2.26}$$

The rotation matrix at each angle (pitch, roll, yaw) is defined as the following matrixes:

$$
\begin{aligned}
R_z &= \begin{bmatrix} \cos(yaw) & -\sin(yaw) & 0 \\ \sin(yaw) & \cos(yaw) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
R_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(pitch) & -\sin(pitch) \\ 0 & \sin(pitch) & \cos(pitch) \end{bmatrix} \\
R_y &= \begin{bmatrix} \cos(roll) & 0 & \sin(roll) \\ 0 & 1 & 0 \\ -\sin(roll) & 0 & \cos(roll) \end{bmatrix}
\end{aligned}
\tag{2.27}
$$

To rotate at its own center $X_{center} = (X, Y, Z)^T$ instead of origin. Ellipsoid first translates to the global origin and starts to rotate at the global origin. Then, translate ellipsoid back to 3D bumblebee centroid.

$$
\begin{aligned}
X_{temp} &= X_{Ellipse} - X_{center} \\
X_{Ellipse} &= (R_z R_x R_y X_{temp}) + X_{center}
\end{aligned}
\tag{2.28}
$$

To minimize error in equation 2.26 with the Nelder-Mead method, an initial condition is needed. But an improper initial condition may lead to a local minimal error instead of a global one. Therefore, to take a reasonable guess for the initial pitch and yaw, adjust the yaw angle first by iteration to target the yaw with minimum angle error but keep the pitch to be 0 in the first step. Then, start with that yaw angle and adjust pitch angle by iteration, minimize final error function $f(\Delta d, \alpha)$ instead of distance error only. Because with non-zero yaw, while changing the pitch angle, it will also affect yaw angle value which leads to a different angle error.

After optimization with initial condition, results shown in Figure 2.19. From the result shown above, clearly, adjust ellipsoid's pitch and yaw angle to fit the projection of ellipsoid well into body contour ellipse is feasible. And in the following section, Artificial bumblebee tests are introduced to evaluate pitch-yaw estimation pipeline performance.

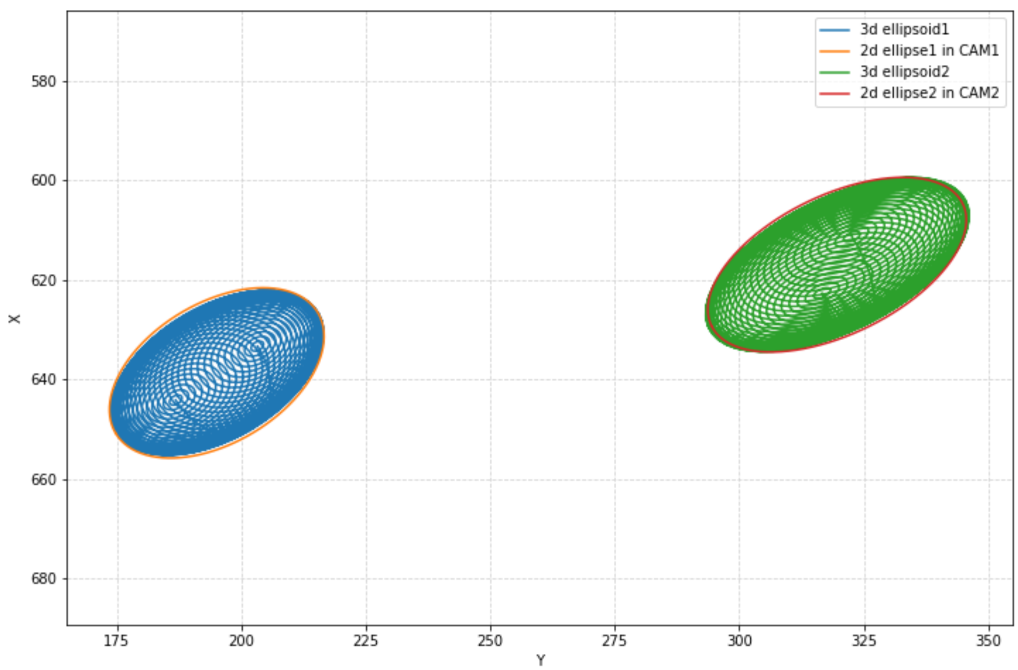

Figure 2.19: Fitting routine results after the Nelder–Mead method

2.4 Artificial bumblebee test set up

Because in the real bumblebee videos, the ground truth like pitch and yaw angles are not easy to obtain. In order to test the performance of the above algorithm pipeline, artificial bumblebee tests have been implemented.

Artificial bumblebee shown in Figure 2.20 are 3D printed from https://www.thingiverse.com/thing:2243456 to mimic real bumble bee body shape. The smaller one has 14mm in width, 34mm in length and 12mm in height.

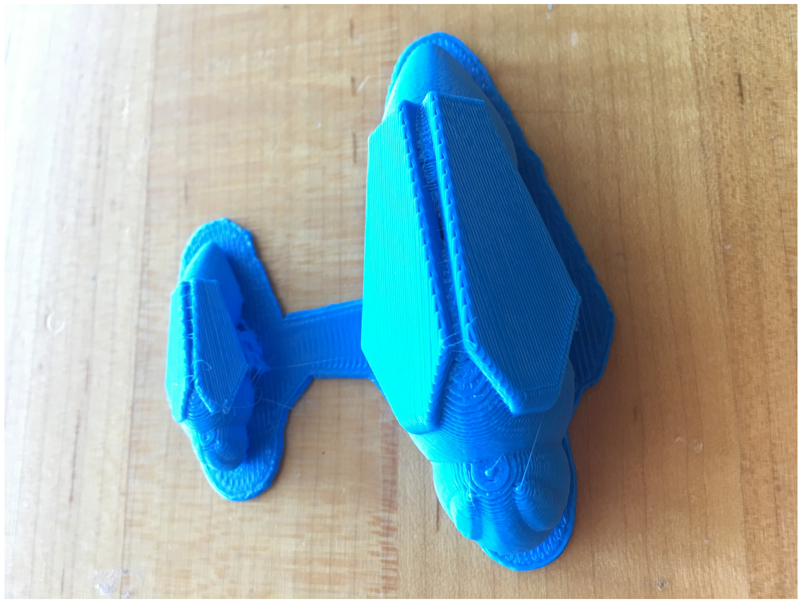

Figure 2.20: 3D printed artificial bumblebee for the algorithm performance test

A plywood disk is glued attached to a monitor hinge mount and is wrapped with a degree circle. Then, 3D printed bumblebee is placed along one of the diameters on the degree circle shown as Figure 2.21 so that this diameter orientation is the same as artificial bumblebee's body orientation in 3D space which can be considered as ground truth.

The local coordinate system is built by translating global coordinate system from global origin to artificial bumblebee centroid Figure 2.21.

With the performance of artificial bumblebee video tests, we could conclude how far the estimated body orientation is away from the ground truth. In the following chapter, the estimated results will be discussed with error analysis to demonstrate how close the estimated values to the ground truth.

Figure 2.21: Set up of artificial bumblebee for body orientation estimation performance test. The yellow coordinate system is the global system and blue and green are local systems.

Chapter 3

Performance Evaluation

As demonstrated before, the body orientation is represented by 3D major principal axis vector of ellipsoid. And the roll cannot be determined only by a 3D vector. Thus, only pitch and yaw will be predicted in the estimation pipeline3.1.

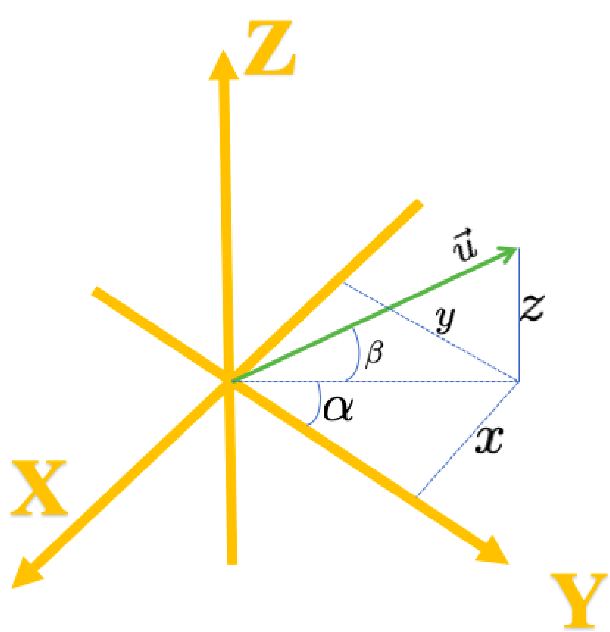

Figure 3.1: Yaw angle is the angle α between the projected vector on XY plane and positive Y-axis and pitch is the angle β between unit vector $\vec{u}$ and XY plane

While manually pushing monitor hinge mount clockwise/counter-clockwise rotates along the global x-axis and triggering high-speed cameras, five videos are recorded and analyzed with different initial pitch and initial yaw angles.

The artificial bumblebee is placed along with one of the diameters on degree circle. By manually clicking two endpoints a_1 and a_2 of this diameter on each camera frame see in Figure 3.2, the diameter 3D orientation can be constructed under triangulation as ground truth which is the same as artificial bumblebee body orientation. The orientation error is defined as $Angle_{estimation} - Angle_{Ground\ Truth}$

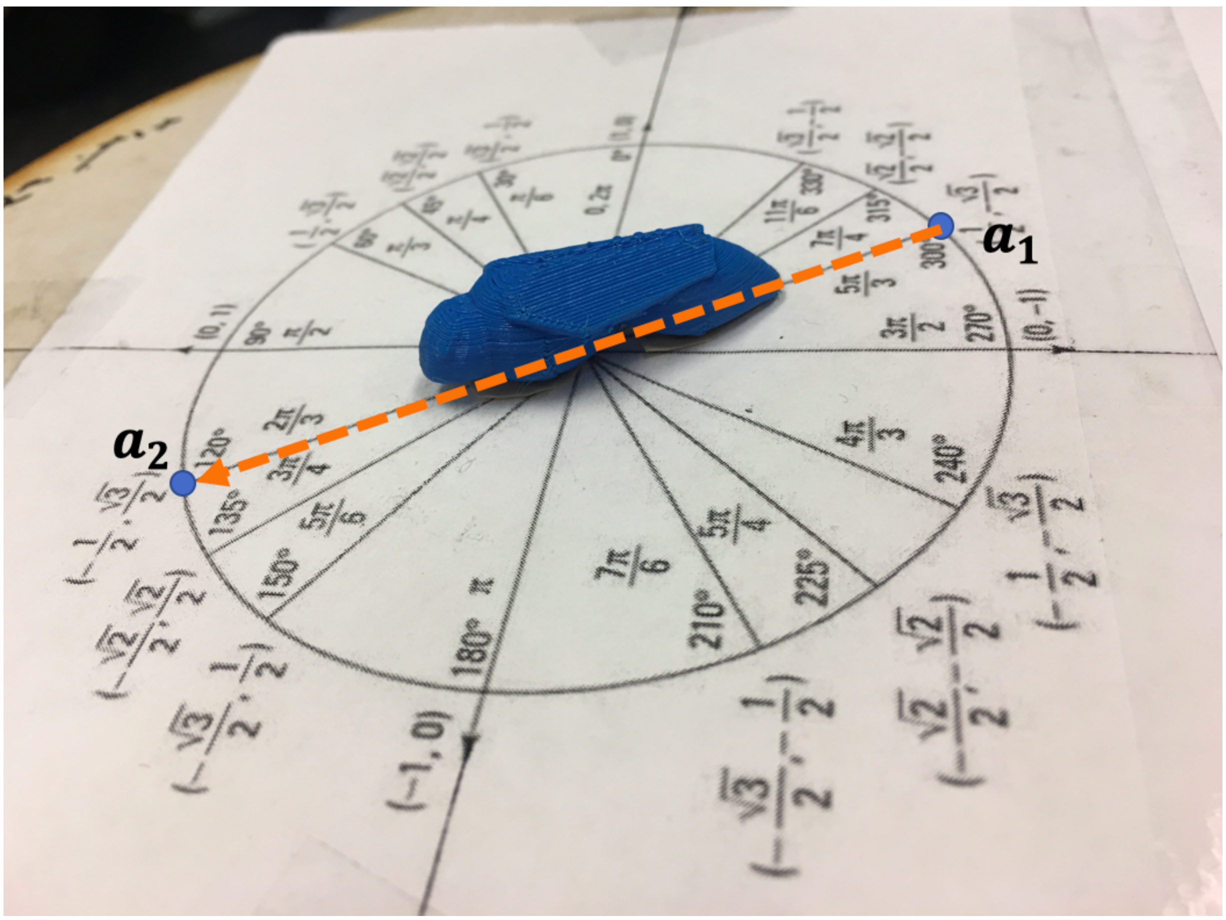

Figure 3.2: Body orientation ground truth vector (Orange arrow) in 3D space

To reconstruct 3D diameter vector's pitch and yaw in 3D space, the following equations 3.1 are developed to reconstruct the pitch and yaw according to Figure 3.1.

$$
\begin{aligned}
u = \frac{V}{|V|} &= \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
yaw &= atan2(y, x) \\
pitch &= arcsin(z)
\end{aligned}
\tag{3.1}
$$

where V is the diameter vector in 3D space and u is its unit 3D vector.

As the results are shown in Figure 3.3 and 3.4, r.m.s error reflects the differences between estimated body orientation and ground truth orientation where its average is 4.41±2.05 in pitch and 3.04±0.90 in yaw among five test videos. Compared to predicted pitch, yaw returns more accurate values especially in $Video2$ and $Video4$. From results shown, typically yaw error is under 5 degrees in most cases. Large pitch error usually exists at a low pitch angle and becomes smaller as pitch goes up Figure 3.5.

In the flowing section, more details about angle error in $video2$ and 4 will be discussed. Because these two videos have larger angle errors than others. Then based on error analysis, some feasible corrections are implemented to reduce error and get closer to the ground truth.

3.1 Error analysis

During both the video 2 and 4, the monitor hinge mount is counter-clockwise rotating along with global x-axis with a low start pitch angle around 5° and 20° respectively. The following parts will discuss potential reasons that lead to angle error and ways to improve the predicted results.

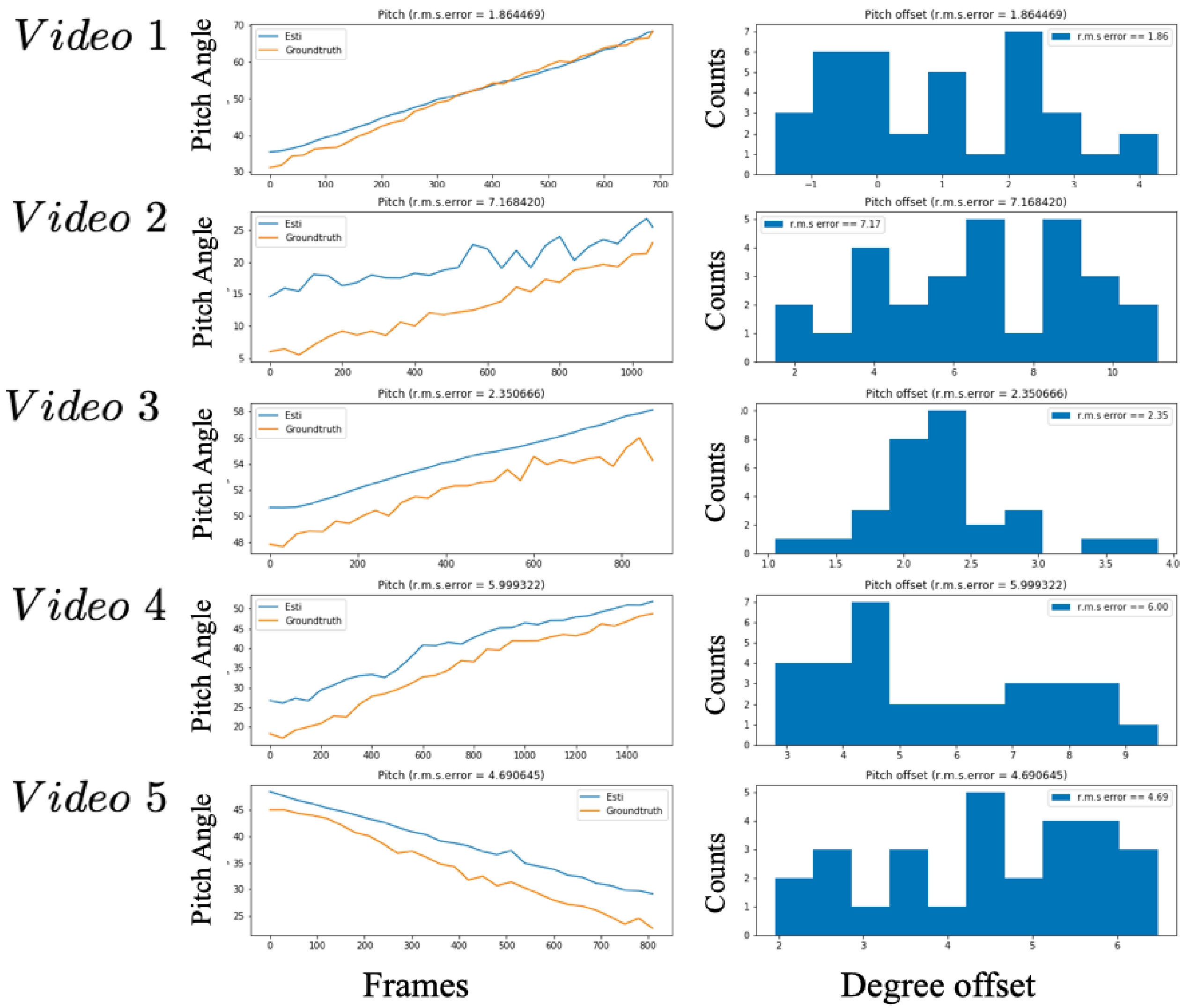

Figure 3.3: The average root mean squared error in pitch among 5 tested videos: 4.41±2.05 deg.

3.1.1 Random error

One possible reason is the random error in fitting routine and 3D reconstruction. Like noise in body contour which introduces error in fitted ellipse geometry. Besides, as manually clicking endpoints of diameters on images, the clicked points could have offset from the ground true points which introduce noise into 3D reconstruction after triangulation.

In the previous video 4 test which has 1500 frames in total, and pitch and yaw are

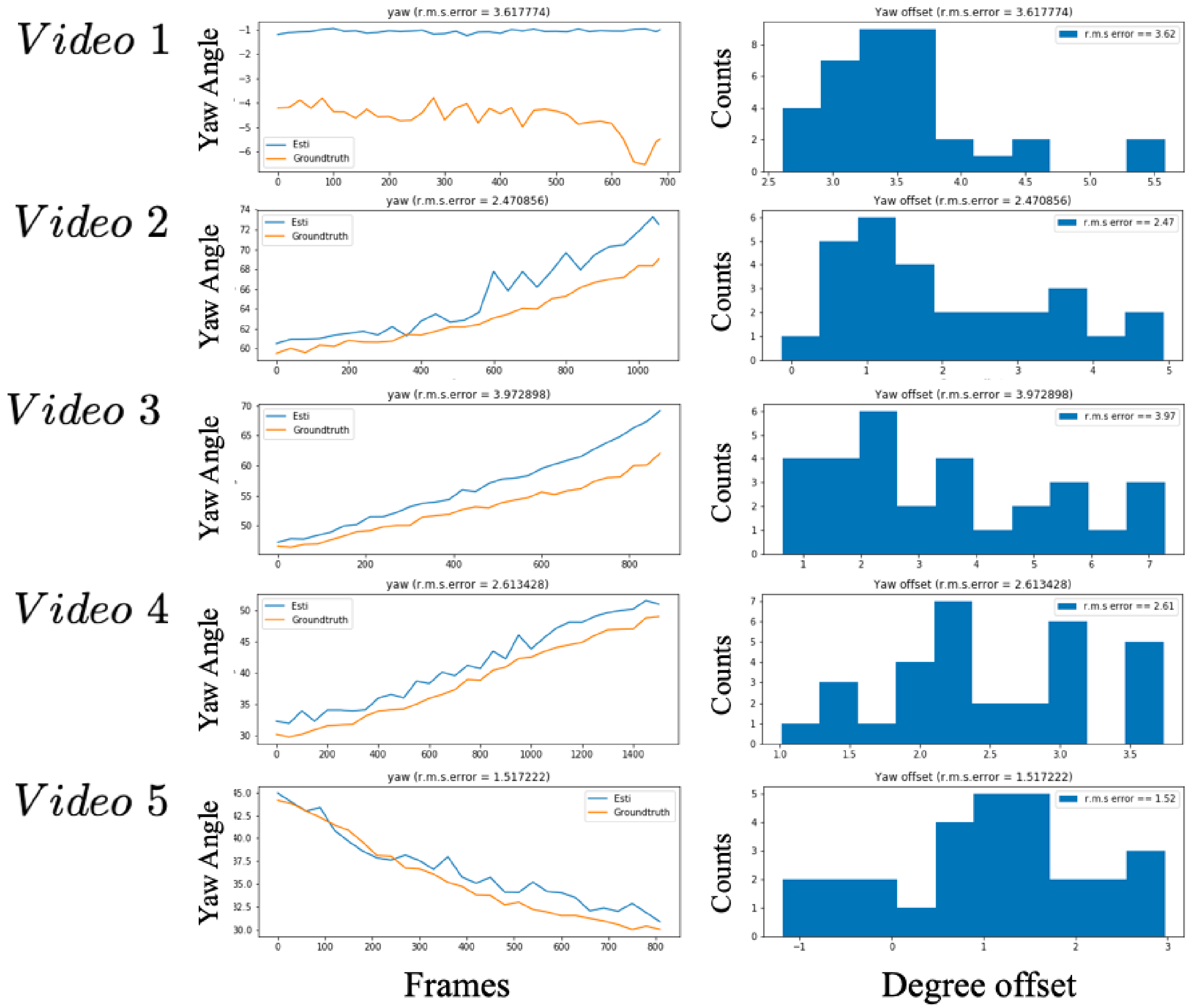

Figure 3.4: The average root mean squared error in yaw among 5 tested videos: 3.04±0.90 deg.

predicted in every 50 frames. If the random error is one of the reasons leads to error in angle estimation. Then the error should be reduced if a smaller step size is applied.

To verify this assumption, *video* 4 have been reanalyzed within 3 kinds of step size $100, 25, 5$. Their 3D reconstructions are shown as the following Figure 3.6.

As step size is decreased from 100 to 5, the r.m.s error in pitch and yaw drop down from 7.28 deg to 4.99 deg and from 3.64 deg to 2.42 deg respectively in Figure 3.7 and 3.8.

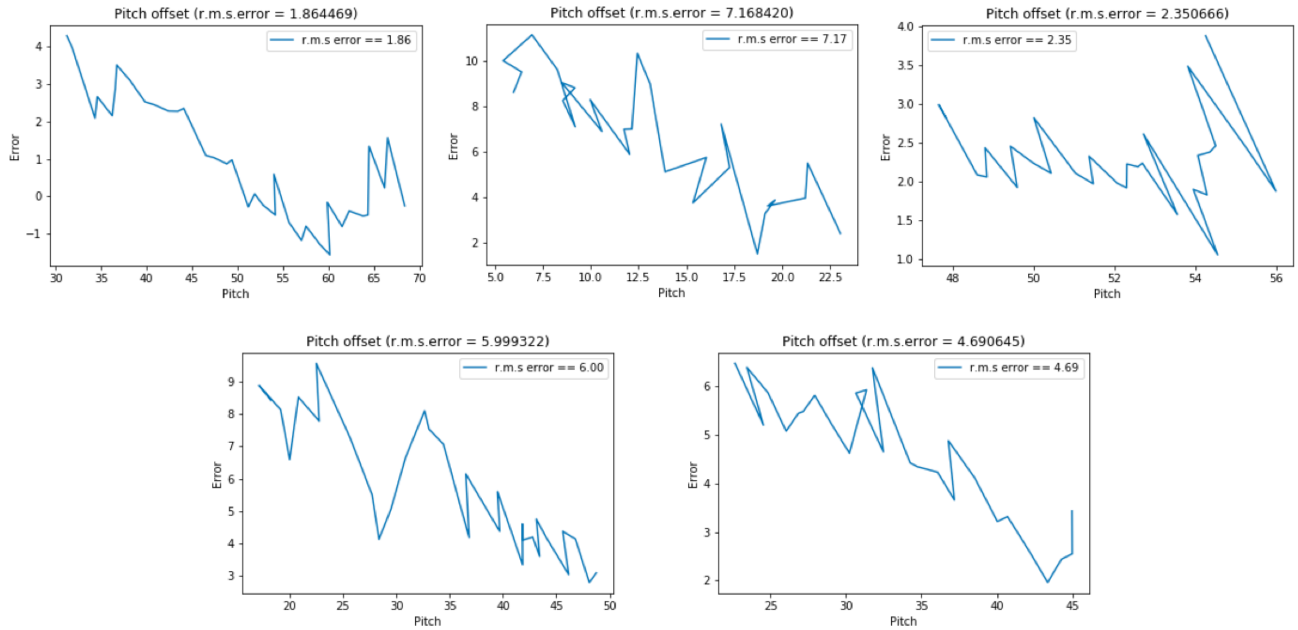

Figure 3.5: Pitch error becomes smaller and smaller as the actual pitch goes up among 5 videos starting from left to right and top to bottom

Therefore, by reducing the step size, it can reduce the random error and let predicted angle converge to the ground truth.

3.1.2 Fitting routine error

Another potential reason is the fitting routine fitEllipse from OpenCV could underestimate the real body length with the least square algorithm. The fitting routine is that body contour's boundary points are computed at first based on contour areas. Then a fitted ellipse is drawn to minimize the sum of squared distance between ellipse and body contour. When artificial bumblebee is at a low pitch position. the tip of the head and abdomen would be ignored to fit the ellipse into major body area which could underestimate the 2D projected bod length shown in Figure 3.9.

Recall Formula 2.23 and 2.24, the fitted ellipse major axis is crucial to minimize

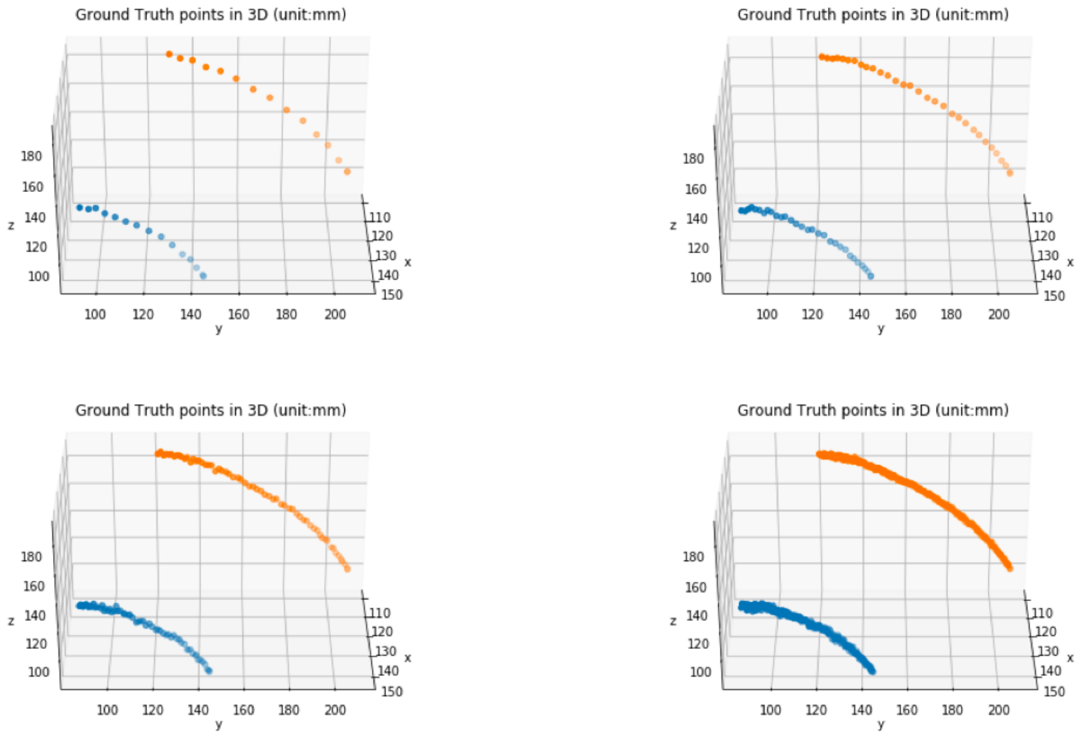

Figure 3.6: The 3D trajectory of diameter's endpoints in Video 4 with step size 100,50,25,5 from right to left and top to bottom.

distance error function and estimate pitch angle. In $Video$ 2 and 4, the artificial bumblebee starting pitch positions are at around 5 and $20\circ$ respectively which lead to a shorter major axis length than body contour length. But to fit this ellipse ($ELPS_1$) into $ELPS_2$, the pitch angle is augmented to shorten $ELPS_2$ major axis length. As monitor hinge mount keeps rotating, the body contour will be rounded and form a more likely ellipse shape Figure 3.9. At that moment, a well-fitted ellipse can be generated without ignoring the tip of the head and abdomen. Therefore, the pitch error is expected to decrease as the pitch goes up which meets the results shown in Figure 3.5.

But, again, to verify assumption, a rectangle is built to bound the body contour with the minimal area with the help of bondingRect function from OpenCV. Although its orientation could be different from body contour, the fitted ellipse returns a good

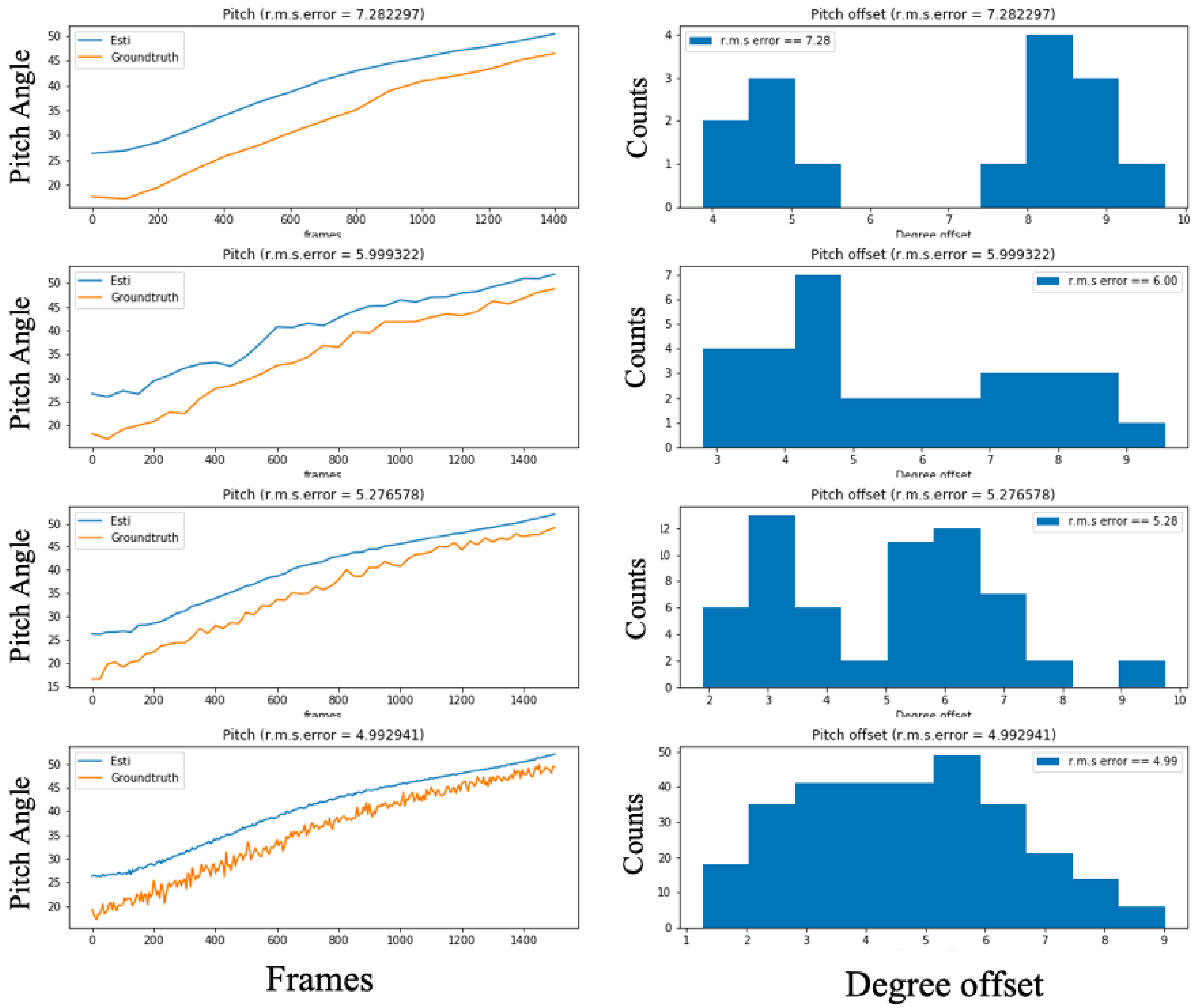

Figure 3.7: Pitch error within different step sizes 100, 50, 25 and 5 from top to bottom has reduced from 7.28 deg to 4.99 deg

estimation of body orientation in 2D. Based on the geometry, 2D projection body length can be reconstructed Figure 3.10.

$$L = a * arccos(\theta_1 - \theta_2) \tag{3.2}$$

where L is the reconstructed body length, a is the rectangle length and θ_1 and θ_2 are the orientation of rectangle and ellipse respectively

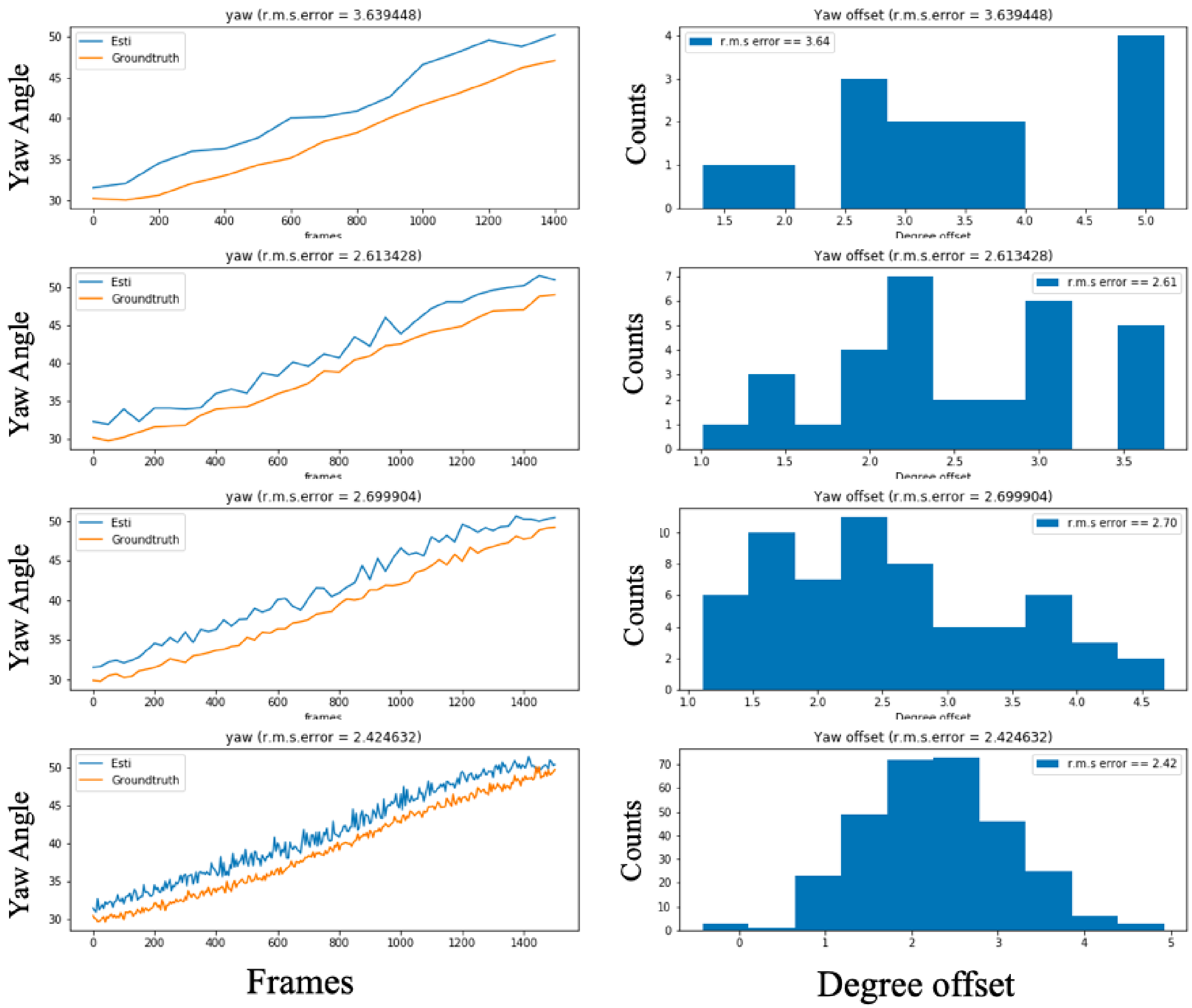

Figure 3.8: Yaw error within different step sizes 100, 50, 25 and 5 from top to bottom has reduced from 3.64 deg to 2.42 deg

After 2D body length reconstruction, the refined results of $Video$ 2 *and* $Video$ 4 are shown in the following Figure 3.11 and 3.12.

Compared with initial results, the r.m.s pitch error is reduced from 7.168 deg to 4.47 deg in $Video$ 2 and from 6.00 deg to 4.55 deg in $Video$ 4. And the r.m.s. yaw angle is reduced from 2.47 deg to 2.16 deg in $Video$ 2 but is slightly increased from 2.61 deg to 2.94 deg in $Video$ 4.

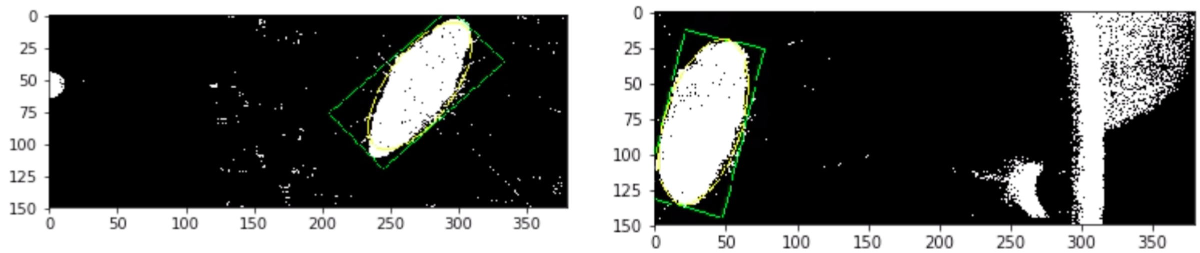

Figure 3.9: The body contour with its fitted ellipse (yellow) and its minimal area fitted rectangle (green). The left figure is the first analyzed frame in $Video$ 4 and the right figure is the last analyzed frame in $Video$ 4.

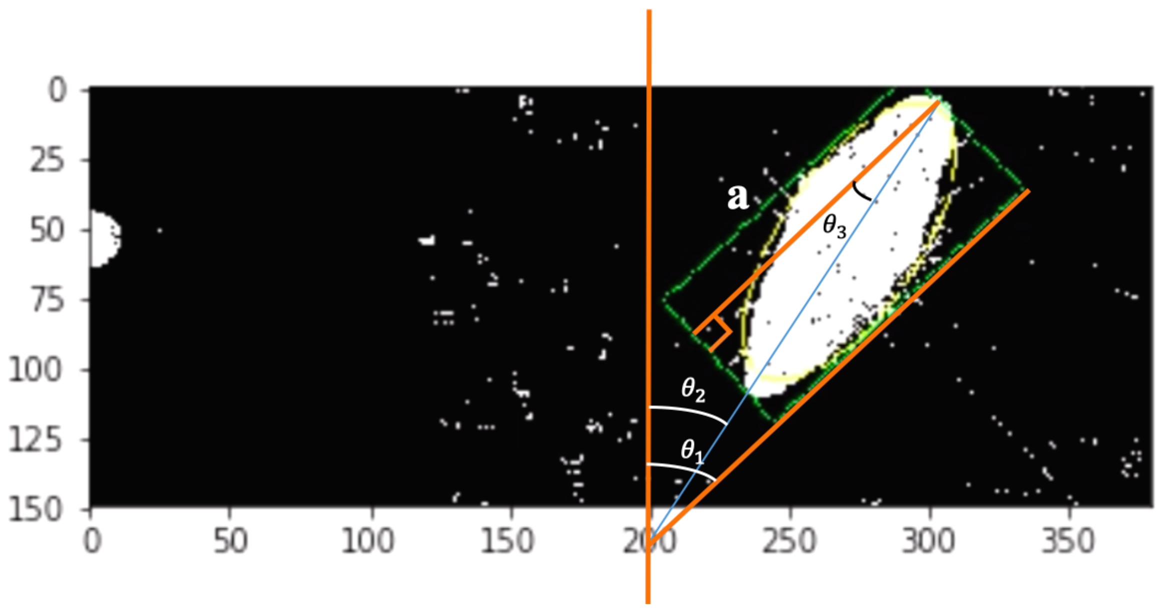

Figure 3.10: Reconstruct the body contour length from minimal bonding area of rectangle

After this correction, the average r.m.s pitch error among five videos is 3.58±1.21 deg and the average r.m.s yaw error among five videos is 2.84±0.90 deg. And this error values could be reduced more if the smaller step size is chosen simultaneously.

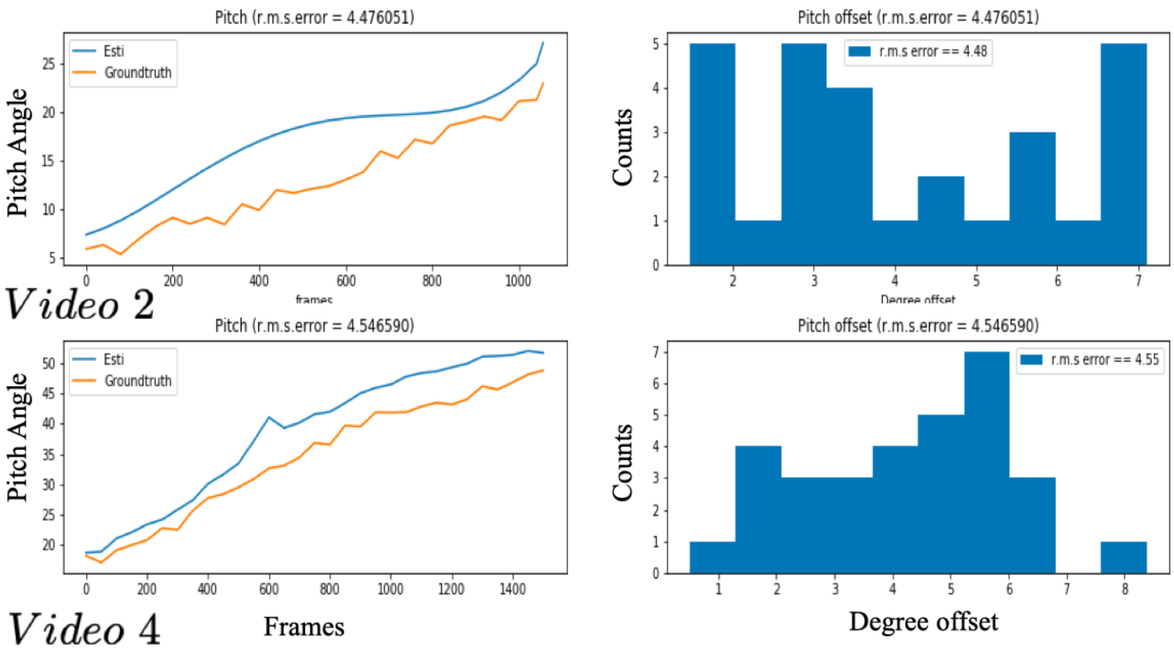

Figure 3.11: Pitch estimation after body length reconstruction of $Video\ 2\ and\ Video\ 4$

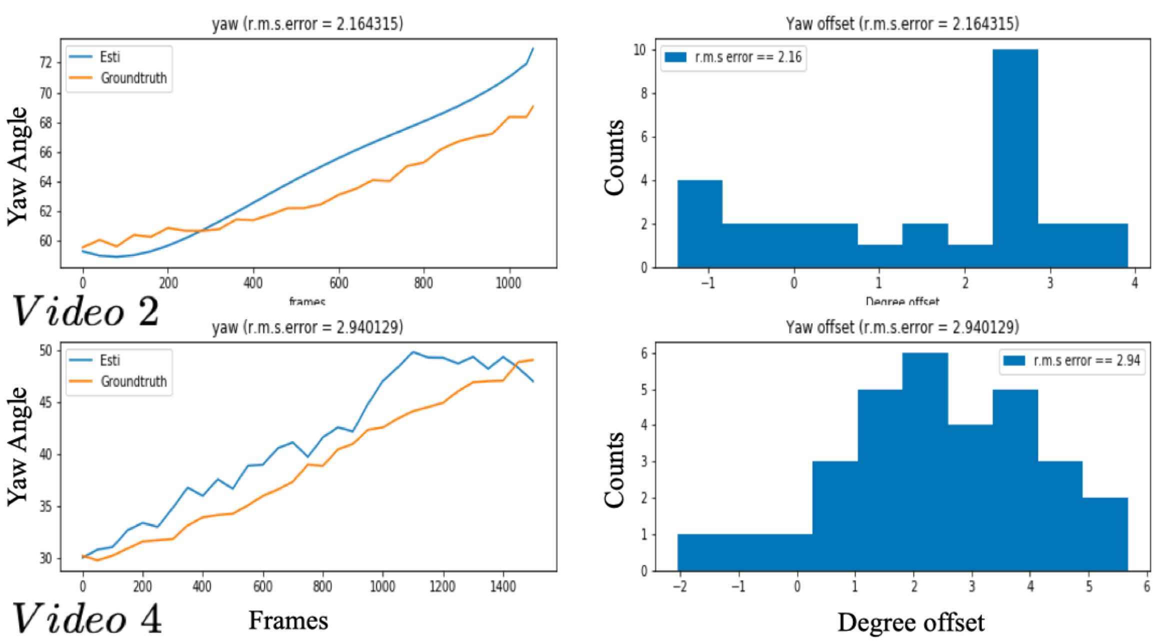

Figure 3.12: Yaw estimation after body length reconstruction of $Video\ 2andVideo\ 4$

Chapter 4

Discussion

We developed a new model-based algorithm for insects' body orientation 3D reconstruction and motion tracking. As demonstrated above, this algorithm shows high performance in accurately tracking body and wings separately with straightforward morphological transformation. Besides, instead of constructing a complex 3D insect body, the insect body is simplified to be a proper ellipsoid which leads to a lucid estimation algorithm structure. The behavior analysis pipeline we developed could return reasonable pitch and yaw under two views only. When the pitch is greater than 30°, pitch and yaw error typically could be under 5 degrees.

4.1 Improvements and future applications

There is still room for improvement. As discussed in error analysis, analyzing more frames in one video could help reduce the random error and reconstructing the fitted ellipse major axis by implementing bondingRect from OpenCV could correct pitch angle. Besides, compared with pitch error, yaw seems to be more accurate. The reason is that yaw estimation depends on fitted ellipse orientation, but pitch estimation depends on fitted ellipse axis length. The fitting routine for body contour provides a better fit in orientation than in length. Also, less accuracy in fitted ellipse axis length results in less accuracy in pitch estimation. Thus, one of the cameras' position may be adjusted to record a proximate side view of the insect so that pitch angle error can also be defined as the orientation difference of two kinds of ellipses in 2D.

Moreover, based on the wing extraction, the wing beat frequency can be computed and the roll angle could also possibly be generated. Because fitted ellipses for wings reflect projected wings orientation in 2D, as shown in Figure 2.15. By analyzing the orientation of the ellipses changing rate and the time period to complete one stroke, wing beat frequency can be estimated. Based on the contour areas of wings shown in Figure 2.14, by applying bondRect and fitEllipse as discussed previously in the Fitting routine error section, two wings' base points in each camera frame could be determined. And then by applying the triangulation method to reconstruct a 3D vector of two base points, the angle rotating along with the body longitudinal axis would be the roll angle.

Based on those improvements discussed above, if we hope to get more accurate pitch angles in the future, two cameras should be located where a proximate side view and top view of the insect can be recorded. If we hope to get roll angle under two views, then two cameras should be placed where two proximate top views of the insect can be recorded. Therefore, a third camera could get involved for improvement of performance.

The body-wings extraction algorithm in this study has been implemented via image thresholding. Thus, this method demands the difference between the brightness of the background and insects' organisms. Thus, any flight insects with transparent wings could meet this requirement. Because the transparent wings cast shadows on the images and its brightness value is higher than the body but lower than the background, this chould lead to a brightness gradient for extraction. Even the flight insect's partial body overlays the wings in some angles of the view, due to the wings' transparent characteristic, the body can still be extracted by setting a lower brightness threshold for later orientation estimation.

Besides, the insect's body consists of a head, thorax, abdomen. During the flight, in most cases, those three parts of the body are aligned along with the body longitudinal axis which can be regarded as a whole and simplified as an ellipsoid. But there still needs to be verified if 3D reconstruction of each part of the body individually improves the algorithm's performance. No matter what kinds of physical models, the way that we defined the error function and optimization routine could provide innovative insights to evaluate how well the physical models' projection fits into the filmed insect's body.

So we have confidence that the pitch-yaw estimation algorithm we developed would perform well on many other kinds of flight insects with transparent wings. Such as drone flies, fruit flies. But this algorithm may fail to separate wings from the insect body due to either the wings have the same brightness as the body, or the wings are not transparent. Like moths, butterflies.

4.2 Conclusion

In this paper, we talked about linear and non-linear camera pose estimation with auto-rejection of outliers and 3D reconstruction under two views. Then, the image processing techniques, including image thresholding, Morphological Transformations and Shape Descriptor are implemented to extract wings and body contours. Also, an innovative algorithm to estimate flight insect's pitch and yaw angle has been demonstrated. The bumblebee's body is simplified as an ellipsoid. By generating the optimal orientation where projections of the ellipsoid best describe the body contours in image frames with minimal fitting error, the current pitch and yaw angle of ellipsoid would be the predicted angles for bumblebee. Therefore, this algorithm can bring a novel perspective into the biological field to reveal insect flight behaviors and control strategies applied during avoidance maneuver.

Bibliography

[Baird et al., 2010] Baird, E., Kornfeldt, T., and Dacke, M. (2010). Minimum viewing angle for visually guided ground speed control in bumblebees. *Journal of Experimental Biology*, 213(10):1625–1632.

[Dudley, 2006] Dudley, R. (2006). Mechanisms and Implications of Animal Flight Maneuverability. *Integrative and Comparative Biology*, 42(1):135–140.

[Fontaine et al., 2009] Fontaine, E. I., Zabala, F., Dickinson, M. H., and Burdick, J. W. (2009). Wing and body motion during flight initiation in Drosophila revealed by automated visual tracking. *Journal of Experimental Biology*, 212(9):1307–1323.

[Fry, 2005] Fry, S. N. (2005). The aerodynamics of hovering flight in Drosophila. *Journal of Experimental Biology*, 208(12):2303–2318.

[Haralick, 2000] Haralick, R. M. (2000). Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. *International Journal of Computer Vision*, 356(1994):331–356.

[Hartley et al., 1996] Hartley, R. I., Sturm, P., K-c, R., and Box, P. O. (1996). Triangulation [CVIU]. *Image (Rochester, N.Y.)*, 68(November 1994):957–966.

[Hedrick, 2008] Hedrick, T. L. (2008). Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinspiration and Biomimetics*, 3(3).

[Lepetit et al., 2009] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166.

[Levenberg, 1943] Levenberg, K. (1943). a Method for the Solution of Certain Non-Linear. *Quarterly of Applied Mathematics*, 1(278):536–538.

[Lin et al., 2014] Lin, H. T., Ros, I. G., and Biewener, A. A. (2014). Through the eyes of a bird: Modelling visually guided obstacle flight. *Journal of the Royal Society Interface*, 11(96).

[Liu and Sun, 2008] Liu, Y. and Sun, M. (2008). Wing kinematics measurement and aerodynamics of hovering droneflies. *Journal of Experimental Biology*, 211(13):2014–2025.

[Marquardt, 1963] Marquardt, D. W. (1963). an algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics 11*, 11(2):431—-441.

[Murray and Road, 1997] Murray, D. W. and Road, P. (1997). The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. *International Journal of Computer Vision*, 24(3):271–300.

[Norberg, 1990] Norberg, U. (1990). *Vertebrate Flight, Mechanics, Physiology, Morphology, Ecology and Evolution*, volume 27.

[Ristroph et al., 2009] Ristroph, L., Berman, G. J., Bergou, A. J., Wang, Z. J., and Cohen, I. (2009). Automated hull reconstruction motion tracking (HRMT) applied to sideways maneuvers of free-flying insects. *Journal of Experimental Biology*, 212(9):1324–1335.

[Serres and Ruffier, 2017] Serres, J. R. and Ruffier, F. (2017). Optic flow-based collision-free strategies: From insects to robots. *Arthropod Structure and Development*, 46(5):703–717.

[Stevenson et al., 1995] Stevenson, Corbo, Baca, and Le (1995). Cage size and flight speed of the tobacco hawkmoth Manduca sexta. *The Journal of experimental biology*, 198(Pt 8):1665–72.

[Sun, 2014] Sun, M. (2014). Insect flight dynamics: Stability and control. *Reviews of Modern Physics*, 86(2):615–646.

[Umeyama, 1991] Umeyama, S. (1991). least squares estimation of transformation parameters between two point pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.