



Lab #1 : Portable Non-invasive Measurement of Drag Coefficients

Jerin Roberts
September 26, 2016

Supervisor: Dr. Patrick Kirchen
Locations: University of British Columbia

Abstract

Measuring the coefficient of drag precisely usually requires a setup which sets an invasive constraint on the object in question. In this experiment we set out to test a method for non-invasive measurement of the drag coefficient for two common sports objects. The method employs object tracking software to calculate the drag coefficient as the object falls to the ground. The average drag coefficients for the shuttlecock and the table tennis ball were found to be 0.949 ± 0.438 and 0.522 ± 0.382 respectively. The viability of the method was tested and found to be accurate compared to external sources. Such a technique could have application in the sports industry provided the precision is improved which will require further research and significant error reduction.

CONTENTS

1. Introduction	4
2. Background	4
2.1. Theory	4
2.2. Measurement	5
3. Methodology	5
3.1. Overview	5
3.2. Procedure	5
3.3. Object Tracking	6
3.4. Calibration	7
3.5. Calculations	8
4. Results	9
4.1. Overview	9
4.2. Sources of Error	10
5. Conclusion	11
A. Appendix	11
A.1. openCV object tracking script	11

LIST OF FIGURES

3.1. Simplified experimental apparatus	6
3.2. A sample output of the openCV program, where the object being tracked is contrasted in red as it falls right to left. Lines are drawn indicating the object is being tracked	6
3.3. Process schematic for video modification using OpenShot video editor	7
4.1. Drag Coefficient as a function of Reynolds (Top) and Standard Deviation as function of Reynolds (Bottom)	9
4.2. Average Drag Coefficient	10

LIST OF TABLES

3.1. Sample data used for calibration constant	8
3.2. Specifications of sports objects	8

1. INTRODUCTION

The force of drag is studied in many engineering fields such as aerodynamics, fluid mechanics and thermal-fluids. Understanding the effects of drag and minimizing its effect can vastly improve performance and efficiency for many systems. For professional sports characterizing the effect of drag on regulation equipment is crucial task. In order for players to compete on equal grounds the sport equipment they use needs to be held to a specific standard. Defining the drag characteristics of a sports ball is one standard that can help keep the game competitive and fair. This is especially critical for sports where drag and lift forces play a key roll in game-play dynamics. Measuring the coefficient of drag precisely usually requires a setup which sets an invasive constraint on the object in question. In this experiment we set out to test a method for non-invasive measurement of the drag coefficient for two common sports objects. Such a technique could have application in the sports industry and could provide a fast portable method for determining if sports objects are within aerodynamic regulation.

2. BACKGROUND

NOMENCLATURE

μ	Dynamic Viscosity $Pa - s$
ρ	Density kg/m^3
A	Cross-Sectional Projected Area m^2
C_D	Drag Coefficient
g	Acceleration due to Gravity in a Vacuum m/s^2
L	Characteristic Linear Dimension m
m	Mass kg
Re	Reynolds Number
v	Velocity m/s

2.1. THEORY

Any object that travels through a fluid will experience an opposing drag force which is related to the objects shape. Drag is generally the result of frictional and viscous forces between the object and fluid. The general expression for force on an object experience due to drag for high velocities is displayed as equation 2.1.

$$F_D = \frac{1}{2} \rho v^2 C_D A \quad (2.1)$$

$$F_g = mg = 9.806m/s^2 \quad (2.2)$$

The term C_D is known as the drag coefficient (dimensionless parameter), F_D is the drag force, ρ is the density of the fluid, v is the speed of the object relative to the fluid, and A is the cross sectional area. For this experiment we are considering the two major forces acting on the sports object as they fall to the ground; gravity (F_g) and drag (F_D). These forces are opposing therefore the measured force will be the sum of gravity and drag as shown in equation 2.3

$$\vec{F}_{object} = \sum \vec{F}_i = (F_D - F_g) \hat{k} \quad (2.3)$$

2.2. MEASUREMENT

There are many methods for measuring the drag coefficient of an object. One of the most common ways is to attach the object to a lift balance in closed wind tunnel. This method ensures the flow is well defined and enables sensitive measurements of drag force for a wide range of flows [Galam, 2009]. However this method is not portable and requires fastening the sport object to a sensor or load cell. Another method for measurement is using a laser to measure the object as it falls through a volume of fluid. This method can also provide details regarding the objects terminal velocity and drag effects for low velocity. However this method again lacks portability as it requires the object to cross a set of laser beams while sinking in a fluid vessel [Cooke, 1999]. In this experiment we plan to test the viability of applying tracking software for portable non-invasive drag coefficient analysis.

3. METHODOLOGY

3.1. OVERVIEW

In order to measure drag we first needed a method to measure the motion of an object falling through the air. By knowing the position of the object relative to time, one will also know the velocity and subsequently the acceleration of the object. The acceleration can be compared to what would be expected if the object were falling in a vacuum, for which any discrepancy between the two will assumed to be caused by drag. The measurement of drag force was found using a video capturing device. Two objects were dropped from varying heights and recorded by a video camera capturing the trajectory to the ground in order to compare the effects of drag.

3.2. PROCEDURE

A GoPro hero 1 camera with a 60fps frame rate was setup a known distance from the screen as shown in figure 3.1. For each run the camera record the flight of the object being dropped. A regulation table tennis ball and a regulation badminton shuttlecock were each dropped 20 times in front of the screen from varying heights and recorded as they fell through the tracking zone. A tracking zone was used in order to reduce the uncertainties associated with

the fish-eye lens of the camera as the effect of the lens was not well characterized. Both the masses and the diameters of the sports objects were measured and recorded.

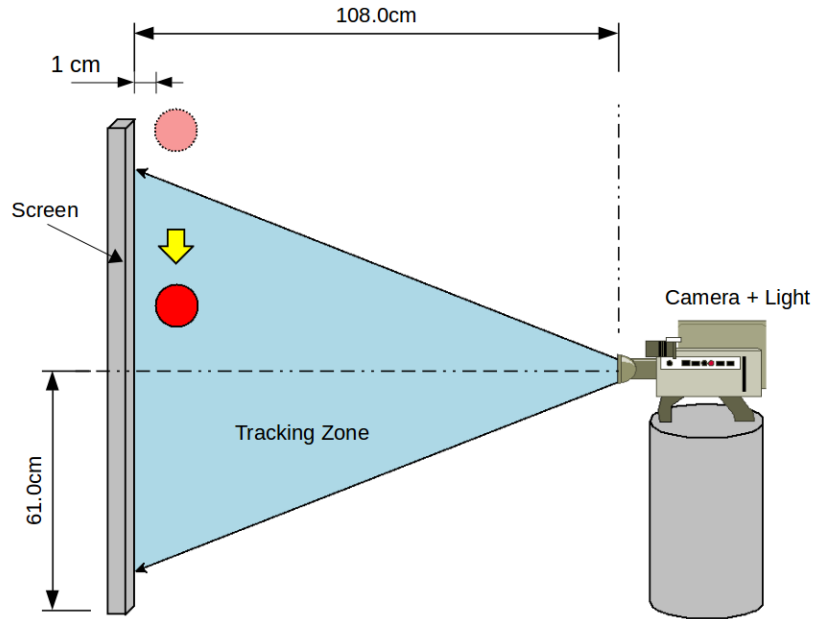


Figure 3.1: Simplified experimental apparatus

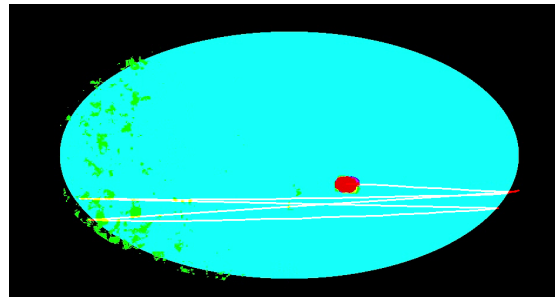


Figure 3.2: A sample output of the openCV program, where the object being tracked is contrasted in red as it falls right to left. Lines are drawn indicating the object is being tracked

3.3. OBJECT TRACKING

In order to measure the distance traveled for each frame, an object tracking program (Appendix A.1) was written making use openCV libraries. The program would track objects of a specified color. For this experiment objects were painted red and allowed to fall in front of a white background greatly contrasting the object. The program filters all colors except for the one specified. Additionally contrasting, brightness and vignette filters were applied to the original

video file in order to emphasize the color of the falling object while filtering out undesirable hues present in the background.

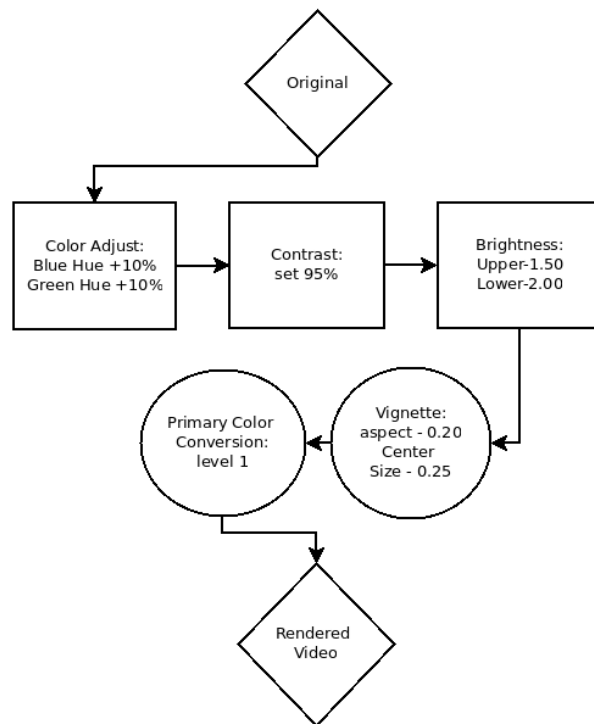


Figure 3.3: Process schematic for video modification using OpenShot video editor

A process schematic for external video rendering is shown in figure 3.3 and example of a filtered frame is displayed in figure 3.2. From here the area and moments that the color occupies are calculated. Using this the program finds the centroid of the contrasted color. If only a large object is detected (beyond noise) a center point of the detected area is recorded for each frame. The program iterates through all frames providing pixel coordinates for the object being tracked. These coordinates are then converted using a calibration constant.

3.4. CALIBRATION

Calibration was needed to convert pixel coordinates into distance displaced by the falling object. Eight small red dots were attached to the white backdrop at $0.03m$ increments. Each dot was recorded individually and its pixel coordinate calculated. The camera's position remained unaltered between recordings.

Pixel Coordinate (x)	Difference (x2-x1)	Average
954.41	n/a	
985.95	31.54	
1016.6	30.65	
1074.46	29.63	
1102.13	27.67	
1129.41	27.28	
1153.72	24.30	0.03m per 28.47 \pm 2.64

Table 3.1: Sample data used for calibration constant

The distance found between each dot was averaged and used as the calibration constant. The pixel data recorded for calibration is displayed in table 3.1 The Calibration constant was found to be $1.05 \times 10^{-3} \pm 9.7 \times 10^{-5}$.

3.5. CALCULATIONS

Sports Object	mass (g)	Area (m^2)
Badminton Shuttlecock	3.51 \pm 0.05	0.002827 \pm 0.00014
Table Tennis Ball	2.73 \pm 0.05	0.001256 \pm 0.00006

Table 3.2: Specifications of sports objects

In order to calculate the drag coefficient we needed to find the force acting on the object due to drag. Since we knew the conventional rate at which an object accelerates in a vacuum, the deceleration due to drag could be known by indirectly measuring the sports balls acceleration (equation 3.1).

$$F_D = m(a_g - a_{object}) \quad (3.1)$$

$$d = At^2 + Bt + C \quad (3.2)$$

Position with respect to time (framerate) was measured and recorded. In order to reduce error amplification a 2nd order polynomial fit, listed as equation 3.2, was applied for each data set where A , B and C are the fit parameters. An expression for the average acceleration was calculated by taking the second derivative of the fit function for distance with respect to time (ie $a = 2A$). The average acceleration for the sports object was referenced against the acceleration due to gravity (vacuum) in order to calculate a value for the force of drag.

$$Re = \frac{\rho v L}{\mu} \quad (3.3)$$

Using the function for velocity ($v = At + B$) and the Force of drag F_D , the drag coefficient C_D was calculated using equation 2.1 and data from table 3.2. The drag coefficients calculated for all 20 runs were then plotted as a function of velocity in the form of Reynolds number Re calculated using equation 3.3. Coefficients found to be less than zero were ignored as this implies faster than gravity acceleration. The error for the calculation was computed by incrementally taking slices for Reynolds numbers and finding the standard deviation of the drag coefficients in that slice. The Standard deviation is then plotted as a function of Reynolds number Re .

4. RESULTS

4.1. OVERVIEW

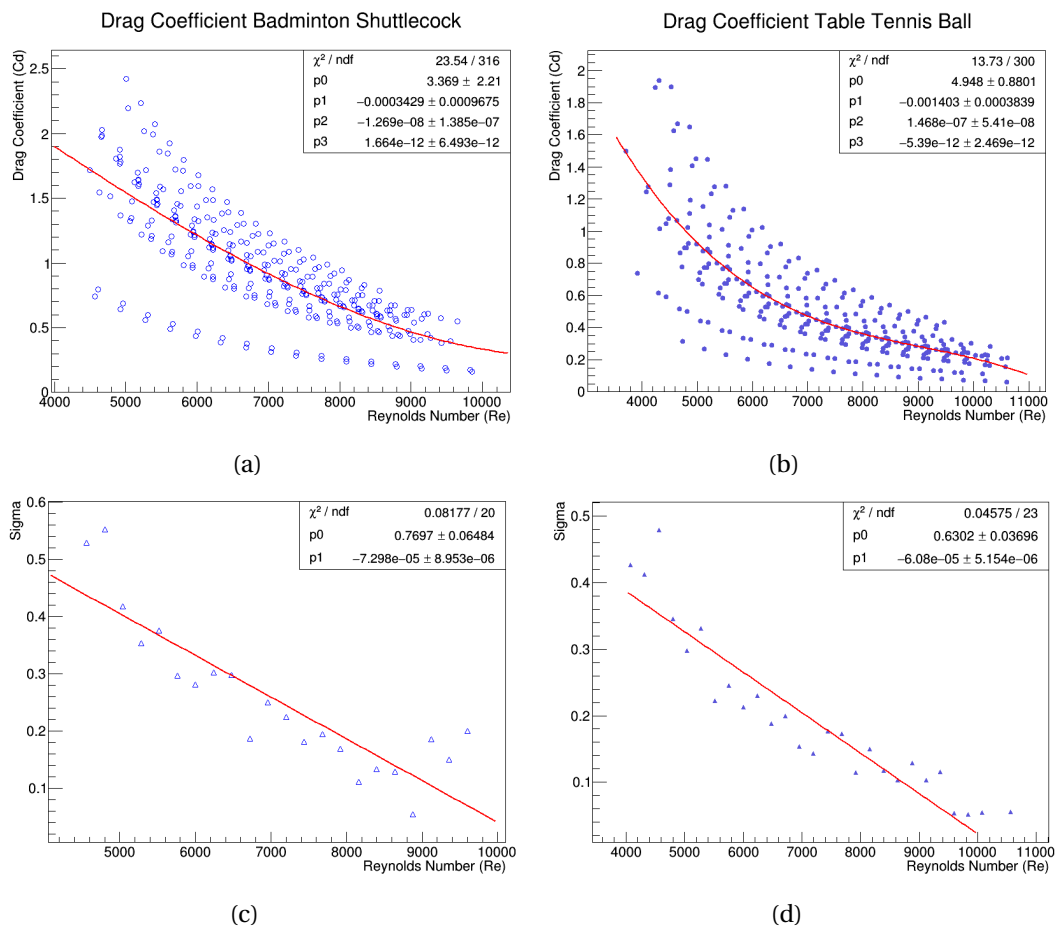


Figure 4.1: Drag Coefficient as a function of Reynolds (Top) and Standard Deviation as function of Reynolds (Bottom)

The drag coefficients of a Badminton Shuttlecock and a Table Tennis Ball were measured for varying velocities and plotted as a function of Reynolds number. Figure 4.1 displays the results of the experiment. Generally the drag coefficient is expressed as an average over measured Reynolds numbers. Figure 4.2 displays the average C_D for both the shuttlecock and the table tennis ball. The average drag coefficient for the shuttlecock and the table tennis ball was found to be 0.949 ± 0.438 and 0.522 ± 0.382 respectively. The average drag coefficient value for a Badminton shuttlecock can vary among studies. Its often found in a range of C_D values from 0.50 to 0.75 for similar Reynolds number range $10^4 - 10^5$ [Cooke, 1999]. Additionally the table tennis ball or spherical object has be known to have C_D values ranging from 0.40 to 0.55 for a Reynolds range of $10^3 - 10^4$ [Cimbala, 2012]. The accuracy of the measured values fall well within these ranges and agree well with current experimental studies. However the precision of the experiment is not within acceptable limits. The size of the standard deviation for slices of Reynolds shows the error of the method is very large (40-50%) . We expect some variance over large Reynolds regions however for small slices this should have be reduced. Based on this method the real value could be found outside standard or well known ranges.

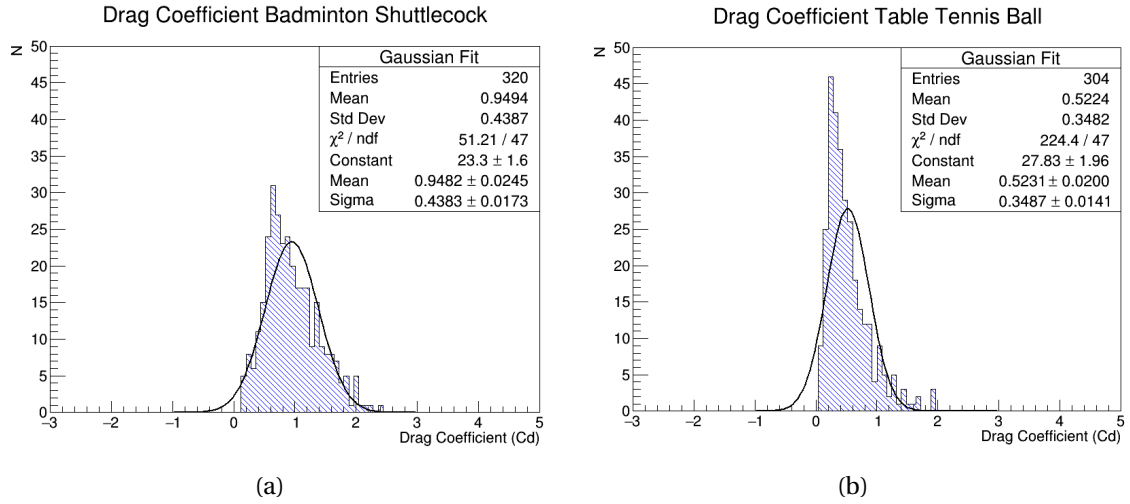


Figure 4.2: Average Drag Coefficient

4.2. SOURCES OF ERROR

There are several sources of error that contribute to this problem. The first is the nature of the measurement method. Since the drag force is not directly measured, we need to infer it by taking two derivatives of the function for measured distance. Any error found in the distance measurement will be inherently amplified in the drag force calculation. Unfortunately, there is no (simple) non-invasive way to measure force. Fitting the displacement function can help this problem but ultimately reduces information known about the system. Therefore, improvements need to be made to the displacement measurement itself. The method of calculating trajectory points based on the centroid of the contrasted area can also cause significant error. As the object moves relative to the light source, its glare changes, distorting

the final shape of the detected area as white objects are not detected. This ultimately causes undesirable shifts to the calculated center point of the tracked area. Using uniform matte colored object and smarter lighting conditions could be used to reduce this error for future tests. Improvements to the video recording system will likely reduce sources of error. The camera used in the experiment had an above average frame rate, however in majority of the frames the objects outline was skewed which causes a shifted center point. This normally indicates the frame rate is too low or the exposure time too great for the speed of the object. A higher resolution camera with a faster capture time formatted for high speed objects would help reduce the errors found in the displacement measurements. Additionally the use of a fish-eye lens was likely a source of error. For future measurements the effect of the lens relative to interpreted displacement should be fully characterized in order to apply a corrective transformation.

5. CONCLUSION

In this experiment we set out to test a method for non-invasive measurement of the drag coefficient for two common sports objects. We were able to test the viability of the method and found it to be accurate compared to external sources. Unfortunately the precision of the method is very poor. Such a technique could have application in the sports industry provided the precision is improved which will require further research. Ultimately based on this analysis this method would provide a poor means for determining the aerodynamic standards on sports equipment. With the majority of source error residing in the software implemented tracking which is the one aspect that makes this method unique, it is unlikely this setup would be suitable for sports related standards policing.

A. APPENDIX

A.1. OPENCV OBJECT TRACKING SCRIPT

```
#include <iostream>
#include <fstream> //printf
#include <unistd.h> //usleep
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <vector>

using namespace cv;
using namespace std;

struct sel{
float xpos;
float ypos;
float time;
```

```

};

struct obj{
vector<sel> eframes;
};

void write_data(string & datafile, obj myset);

int main( int argc, char** argv )
{
    string myfile = "tmp";
    obj myset;

    VideoCapture cap("video/ping1.mp4"); //video from file

    if ( !cap.isOpened() ) // if fails
    {
        cout << "Cannot open the web cam" << endl;
        return -1;
    }

    namedWindow("Control", CV_WINDOW_AUTOSIZE); //create a window

    int iLowH = 0;
    int iHighH = 179;
    int iLowS = 255;
    int iHighS = 255;
    int iLowV = 189;
    int iHighV = 255;

    //Create trackbars in "Control" window
    createTrackbar("LowH", "Control", &iLowH, 179); //Hue (0 - 179)
    createTrackbar("HighH", "Control", &iHighH, 179);
    createTrackbar("LowS", "Control", &iLowS, 255); //Saturation (0 - 255)
    createTrackbar("HighS", "Control", &iHighS, 255);
    createTrackbar("LowV", "Control", &iLowV, 255); //Value (0 - 255)
    createTrackbar("HighV", "Control", &iHighV, 255);

    float iLastX = -1;
    float iLastY = -1;

    //Capture a temporary image from the camera
    Mat imgTmp;
    cap.read(imgTmp);

    //Create a black image with the size as the camera output
    Mat imgLines = Mat::zeros( imgTmp.size(), CV_8UC3 );
    int frame = 0;

```

```

while (true)
{
    //usleep(100000);
    ++frame;
    Mat imgOriginal;
    bool bSuccess = cap.read(imgOriginal); // read a new frame from video

    if (!bSuccess)
    {
        cout << "Cannot read a frame from video stream" << endl;
        break;
    }

    Mat imgHSV;
    cvtColor(imgOriginal, imgHSV, COLOR_BGR2HSV); //Convert the captured frame
    from BGR to HSV
    Mat imgThresholded;
    inRange(imgHSV, Scalar(iLowH, iLowS, iLowV), Scalar(iHighH, iHighS, iHighV),
        imgThresholded); //Threshold the image

    //removes small objects from the foreground
    erode(imgThresholded, imgThresholded, getStructuringElement(MORPH_ELLIPSE,
        Size(5, 5)) );
    dilate( imgThresholded, imgThresholded, getStructuringElement(MORPH_ELLIPSE,
        Size(5, 5)) );

    //remove small holes from the foreground
    dilate( imgThresholded, imgThresholded, getStructuringElement(MORPH_ELLIPSE,
        Size(5, 5)) );
    erode(imgThresholded, imgThresholded, getStructuringElement(MORPH_ELLIPSE,
        Size(5, 5)) );

    //Calculate the moments of the thresholded image
    Moments oMoments = moments(imgThresholded);

    double dM01 = oMoments.m01;
    double dM10 = oMoments.m10;
    double dArea = oMoments.m00;
    float posX = 0;
    float posY = 0;
    float vX = 0;
    float vY = 0;

    // if the area <= 1000, rest is noise
    if (dArea > 1000)
    {
        //calculate the position of the ball
        sel temppoint;
        posX = dM10 / dArea;
        posY = dM01 / dArea;
        temppoint.xpos = posX;
    }
}

```

```

    temppoint.ypos = posY;
    temppoint.time = frame;

    printf("Xpos: %f Ypos: %f frame: %d\n", posX, posY, frame);

    myset.eframes.push_back(temppoint);
    if (iLastX >= 0 && iLastY >= 0 && posX >= 0 && posY >= 0)
    {
        //Draw a red line from the previous point to the current point
        line(imgLines, Point(posX, posY), Point(iLastX, iLastY), Scalar(0,0,255),
            2);
    }
    iLastX = posX;
    iLastY = posY;
}

imshow("Thresholded Image", imgThresholded); //show the thresholded image

imgOriginal = imgOriginal + imgLines;
imshow("Original", imgOriginal); //show the original image

    if (waitKey(30) == 27) //wait for 'esc' key press for 30ms. If 'esc'
        key is pressed, break loop
    {
        cout << "esc key is pressed by user" << endl;
        break;
    }
}
write_data(myfile, myset);
return 0;
}

void write_data(string & datafile, obj myset)
{
    sel temp;
    std::ofstream myfile;
    datafile.append(".txt");
    const char * c;
    c = datafile.c_str();
    myfile.open(c, fstream::out);

    for(int i = 0; i < myset.eframes.size(); ++i)
    {
        sel temp;
        temp = myset.eframes.at(i);
        float posX = temp.xpos;
        float posY = temp.ypos;
        float frame = temp.time;

```

```
myfile << "data " << posX << " " << posY << " " << frame << "\n";  
printf("writing Xpos: %f Ypos: %f frame: %f\n", posX, posY, frame);  
}  
myfile.close();  
}
```

REFERENCES

- [Galam, 2009] Firoz Alam, Harun Chowdhury, C. Theppadungporn and Aleksandar Subic (2009) "A study of Badminton shuttlecock aerodynamics", RMIT University
- [Cooke, 1999] Alison J. Cooke (1999) "Shuttlecock Aerodynamics", Department of Engineering, Cambridge University
- [Cimbala, 2012] J.M. Cimbala (1999) "Drag on Spheres", Department of Engineering, Penn State University