FYS2150
Lab Report: Drag

Nicholas Karlsen

April 29, 2018

Abstract

A study on the flow of an assortment of spheres in a fluid and the use of image processing to determine the terminal velocity.

1 Introduction

This report contains the description and analysis of data collected in the lab 21.03.2018 concerning the flow of several spherical objects in a large range of different sizes and densities. The balls were immersed in fluid, dropped and filmed. Post-lab, the raw footage was then processed using a Python script in order to quantify the motion of the spheres. This

2 Theory

2.1 Image processing

$$Y_{i,j} = 0.2126 \cdot \mathrm{I}_{i,j,1} + 0.7152 \cdot \mathrm{I}_{i,j,2} + 0.0722 \cdot \mathrm{I}_{i,j,3} \tag{1}$$

An image can be represented by an $M \times N$ array, where each entry, $\mathrm{I}_{i,j}$ is a list of varying length depending on the color space of the image, consisting of numbers from 0 to 256, signifying the intensity of each color. RGB 3 entries and gray scale 1 entry. Conversion from RGB to gray scale is calculated by Eqn. 1 where $\mathrm{I}_{i,j,\ldots}$ denotes array elements of the input RGB image and $\mathrm{Y}_{i,j}$ the output gray scale image. [5]
A gray scale image can be further simplified by converting it to a binary black or white image. i.e, an image consisting only of black or white pixels. Such an image can be created by Eqn. 2 where $X_{i,j}$ denotes the array element of the output binary image and T the threshold, a number between 1 and 256.

$$X_{i,j} = \begin{cases} Y_{i,j} \geq T & 256 \\ Y_{i,j} < T & 0 \end{cases} \tag{2}$$

The properties of a binary image can then be determined by an algorithm such as regionprops() which determines the properties of shapes present in the image. For the purposes of analyzing a video, and finding the position of one particular shape in each frame of that video, it is important that the binary image only contains one shape and minimal noise, else regionprops() may not work as desired.

In order to achieve this, two methods in particular are easy to implement. 1) If the object moves in an "open" area, cropping the image may suffice as a way to filter out anything undesirable. This is the most efficient method as far as computation time is concerned. It does however limit the possibility of completely automating the analysis of a large quantity of videos. 2) If the background is entirely static, or almost. The background can be filtered out by multiplying each frame of the video by an inverted image of the background divided by 256. Such an inverted image will have value 0 where the static objects are located, and 1 everywhere else.

3 Experimental Procedure

3.1 Video capture

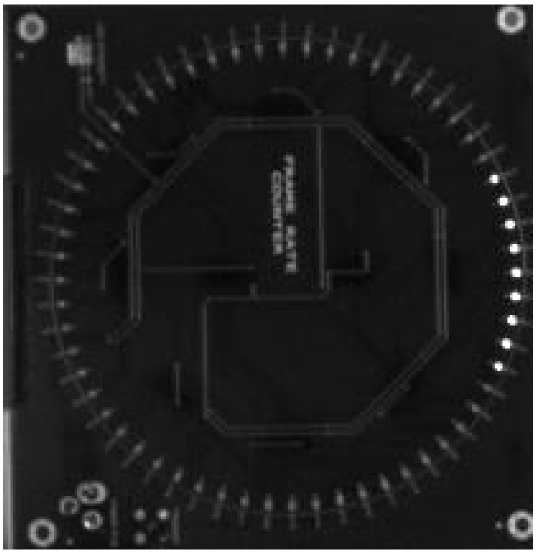


Figure 1: Signal used to determine the FPS of the camera

In order to capture the motion of the balls, a USB video camera was connected to a computer running uEye cockpit[4], a program which allowed us to change the settings of the camera, as well as record video files. The image was set to grayscale and the contrast, exposure, gain etc was changed such that the color histogram in uEye would show two significant, separated peaks such that it would be easy to process the image using the method described in section 2.1.

Next, the error of the set FPS of the camera had to be determined. This was done by connecting a series of LEDs in a circle (see Fig. 1) to a signal generator. The light emitted would "circle" at a rate which could be changed using the signal generator. By adjusting the rate such that the emitted light would seem stationary when observed through the video feed in uEye, the true FPS of the camera could be determined. In our test, we set the camera to film at 60 FPS and the rate was adjusted untill the period of the LED was synced up with the camera at 3.60047 KHz, the equivalent of 60.00783 periods per second.

In order to get the conversion factor between pixels and meters, a still picture was taken of the tubes with a known length in frame, at a distance roughly equal to where the balls would travel. This had to be done for each experimental setup. For our known lengths, we used a meter rule and a 30 cm ruler for the large and small scale experiments respectively. See Fig. 4.

(a) Large tube (b) Small tube

Figure 2: Cropped images used to find pixel to meter ratio for both of the tubes. (Scaled down in this document)

3.2 Equipment

We had two similar experimental setups. One large scale, and one small scale. the large scale consisted of a roughly 2.5-3m tall[1] transparrent tube, and the small scale a 30-40cm transparent measuring tube. Both of the tubes were filled with oil (Shell Tellus S2 M 68 Industrial Hydraulic Fluid[3]).

The set of balls which were to be dropped in the large scale setup are depicted in Fig. 3. Additionally, there were two smaller balls which were dropped in the small scale setup. The mass of the balls was measured using a digital scale with a precision of 0.01g

[1] We did not measure the height of the tube, but it was fairly close to the ceiling.

[2], except for the two smallest balls, for which the mass was given by the lab instructor. The diameter of the balls was measured using a Vernier Caliper with a precision of 0.01mm, again, except for the two smallest balls, which diameter was provided by the lab instructor.

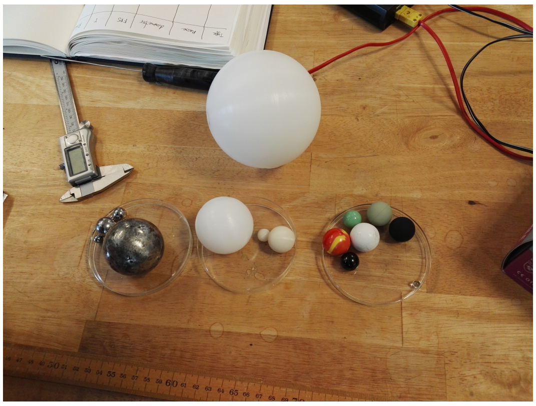


Figure 3: Most of the balls used in the experiment, excluding the ones labeled small 1 and 2.

3.3 Procedure

The experiment was performed by 3 people performing 3 different, simultaneous tasks. It can technically be performed by one person, but due to the nature of the experiment, it is not recommended, as it is not practical to make repeated readings if something goes wrong.

The camera was pointed towards the large-scale tube, and prepared as described in section 3.1 between each reading. It was important that the lighting in the room was fairly constant for the video capture, so curtains were put in front of the windows. The camera was kept at 100 FPS for every experiment. For some of the balls with relatively low terminal velocity, the FPS could have been lowered to save space without making any significant sacrifices to the accuracy, but as we did not have any predicted values for the terminal velocity of the individual balls, we opted to keep it at 100FPS for every single experiment.

Before each experiment, the temperature of the oil was measured using an IR-thermometer with a precision of $\pm 2°C$ [1]. Which is used to determine the viscosity of the oil.

For each experiment, a ball was picked up by hand and brought to the top of the tube by ladder. The ball was gripped tightly before carefully submerging it in the oil. After having ensured the person operating the camera software was ready to record, the ball was dropped. It was critical that the ball was dropped in a certain way, otherwise it gained too much horizontal velocity, which at worst could cause it to hit the side of the tube halfway through[2] the tube. In order to avoid this, it is important to separate

[2]Which happened in our first attempt

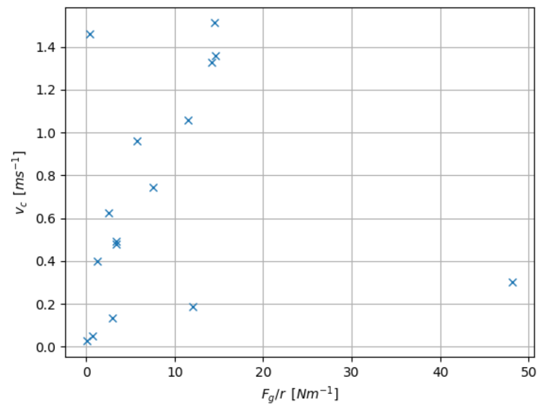
the fingers gripping the ball at the same time when dropping it, instead of letting it slide out of the hand. Sliding it out, leads it to spin, which in turn may lead it to move horizontally.

The same procedure was performed for the small-scale experiments.

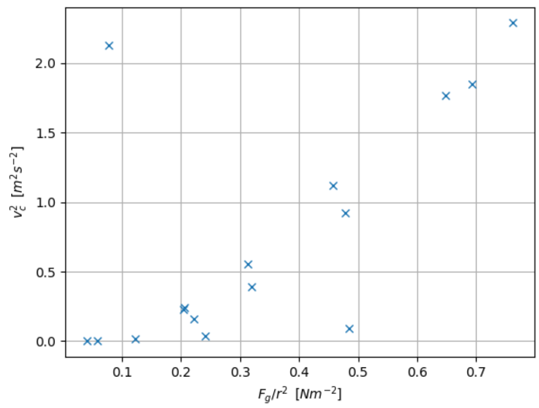
4 Results

Table 1: Results

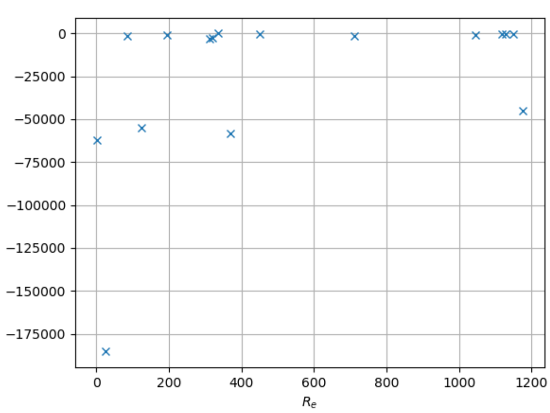
Type	Mass [g]	Diameter [mm]	Terminal Velocity [ms^{-1}]	FPS	T [°C]	Filename
Metal	502.76	48.98	N/A	100	22.7	A1.avi
Metal	28.13	19.02	1.5122	100	22.8	A2.avi
Metal	6.99	11.97	0.9588	100	22.6	A3.avi
Metal	2.08	7.99	0.6238	100	22.6	A4.avi
Metal	0.68	5.48	0.3980	100	22.5	A5.avi
Metal	0.10	2.98	N/A	100	22.6	A6.avi
Plastic	488.41	99.4	0.3006	100	22.5	B1.avi
Plastic	61.56	50.02	0.1873	100	22.5	B2.avi
Plastic	7.12	23.89	0.1337	100	22.6	B3.avi
Plastic	0.87	12.06	0.0517	100	22.6	B4.avi
White	29.74	25.24	1.0572	100	22.6	C1.avi
BigRed	18.44	24.01	0.7455	100	22.3	C2.avi
BigGreen	31.60	21.86	1.3288	100	22.4	C3.avi
BigBlack	31.42	21.08	1.3592	100	22.5	C4.avi
SmallBlack	5.67	16.45	0.4934	100	22.5	C5.avi
SmallGreen	5.60	16.38	0.4783	100	22.3	C6.avi
Glass	0.27	5.81	1.4592	100	22.3	C7.avi
Small1	4.1e-3	1.0	0.0257	100	23.7	D1.avi
Small2	12.0e-3	1.59	N/A	100	23.7	D2.avi



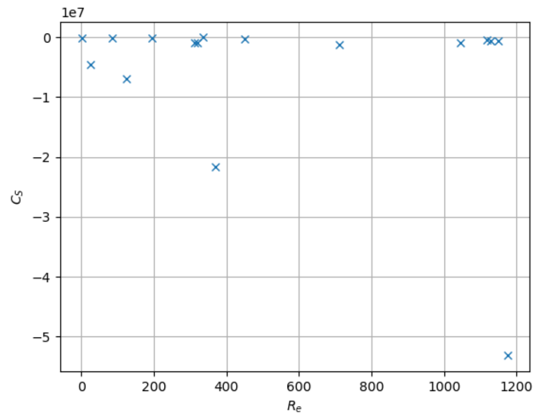
(a) v_c against F_g/r



(b) v_c^2 against F_g/r^2



(c) C_R against R_e



(d) C_S against R_e

Figure 4: Data sets

5 Discussion

6 Conclusion

References

[1] Fluke 62 user manual.

[2] ProScale XC user manual.

[3] Shell tellus s2 m 68 technical data sheet.

[4] uEye cockpit - IDS imaging development systems GmbH.

[5] Wikipedia contributors. Grayscale — wikipedia, the free encyclopedia. Page Version ID: 825417562.