**Particle Shape Characterization using Fractal Analysis**

*Internship Report*

by

**Gautam Kumaran**

Under the guidance of

**Dr. Mousumi Mukherjee**

A picture containing graphics, font, graphic design, screenshot

Description automatically generated

**SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY MANDI**

**Mandi-175005, India**

**July-2023**

**ACKNOWLEDGMENTS**

I would like to thank Mr. Madhu Sudan for his help and guidance throughout the course of my internship in work and non-work-related matters.

**Introduction**

Barrett (1980) proposed that the shape of a particle can be described by three properties, namely overall form, angularity, and roughness. Overall form carries information on the proportions of the particle at the macro-scale, i.e., on how isometric or elongated the particle is; angularity accounts for local features of the particle at the meso-scale; roughness describes the texture of the particle surface at the micro-scale. The aim of this project is to implement a simple and effective method, based on the robust mathematical framework of fractal analysis, to characterise the morphology of soil particles in terms of three new quantitative descriptors that can be associated systematically to the above three properties.

**Methodology**

1. **Image Processing**

The first step to analysing the particle in the image is to prepare the image. We must isolate the particle in the image and extract its boundary. Only once that is complete can we start the fractal analysis.

**1.1. Gamma correction**

The first step is to make the image grayscale and make it brighter to help with future steps. We do this by using a method called gamma correction. The Gamma correction method makes an image brighter which makes it easier to differentiate between the foreground and background.

The method is done by rescaling the pixel intensity from between 0 and 255 to 0 and 1. After rescaling the pixels, raise it to the power of 1/gamma where if gamma is lower it makes the image darker and if gamma is higher, it makes it brighter. Then we rescale it back up to 0 to 255

A close-up of several rocks

Description automatically generated A close-up of several rocks

Description automatically generated

(a) (b)

Fig 1: (a) Original Image, (b) Gamma Corrected Image (Gamma = 2.2)

**1.2. Otsu’s Method**

The Otsu’s method determines how well a certain threshold value separates the foreground from the background using the following formula.

- (1)

where,

* = (number of pixels in background)/total pixels
* = (number of pixels in foreground)/total pixels
* = (sum for all I from 0 to assumed threshold (number of pixels in intensity I \* I))/total pixels in background
* = (sum for all I from assumed threshold to 255 (number of pixels in intensity I \* I))/total pixels in foreground

Let’s take the following example.

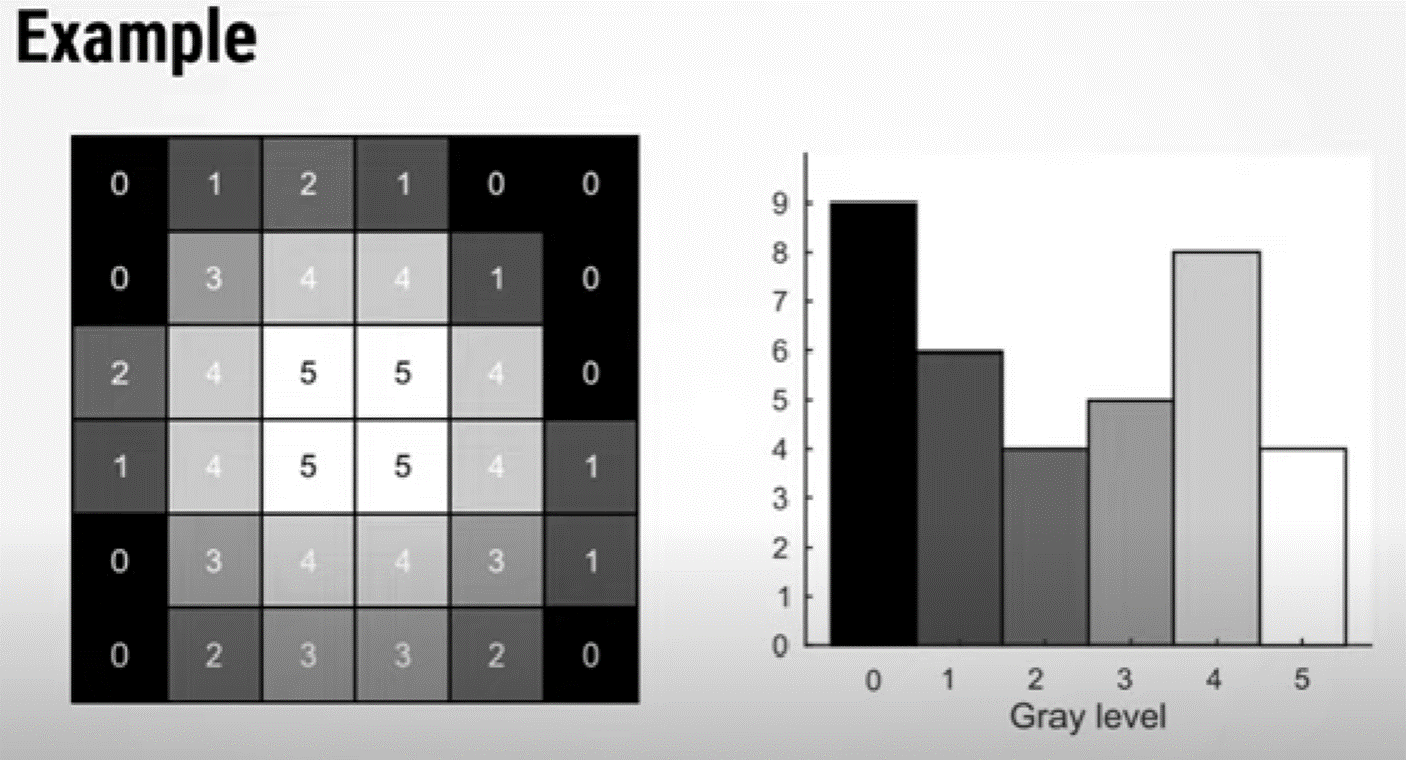


Fig 2: Example of image with 6 different intensities with histogram showing the frequencies.

There are 6 possible threshold values. Hence, we calculate the 6 possible values using the first formula which gives us the following table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Intensities** | **0** | **1** | **2** | **3** | **4** | **5** |
|  | 0 | 0.25 | 0.42 | 0.53 | 0.67 | 0.89 |
|  | 0 | 0 | 0.40 | 0.74 | 1.21 | 1.91 |
|  | 1 | 0.75 | 0.58 | 0.47 | 0.33 | 0.11 |
|  | 2.25 | 3.00 | 3.57 | 3.94 | 4.33 | 5.00 |
|  | 0 | 1.69 | 2.44 | 2.56 | 2.17 | 0.95 |

Table 1: Displaying the variance for different threshold values

From this table we can see that the threshold value 3 gives the best separation. So that is our threshold value for the next step.

**1.3 Image Binarization**

A diagram of a flowchart

Description automatically generated

Fig 3: Flowchart for binarization method

Now that we have our threshold value, we use the above algorithm to classify each pixel to the foreground and background. Once classified we rewrite the image to make all foreground pixels white and all background pixels black. Hence converting the image to a binary image.

A white and black image

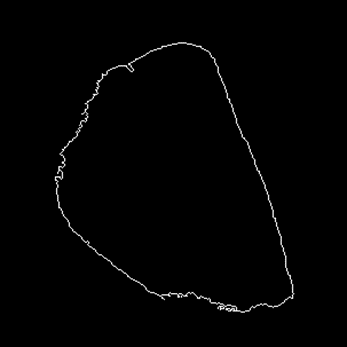
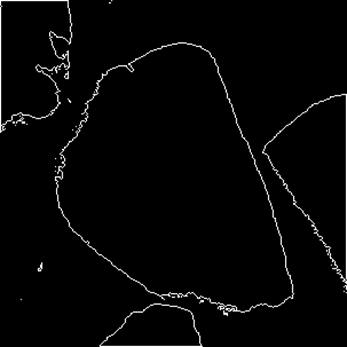
Description automatically generated

(a) (b)

Fig 4: (a) Gamma Corrected Image, (b) Binarized image

**1.4 Isolating the Boundary.**

The next step is to isolate the boundary of the particle from the binarized image. To do this, we first find all the boundaries of all particles in the image(left). We then isolate the biggest boundary since the biggest one is the boundary of the particle we want to observe(right).



(a) (b)

Fig 5: (a) All boundaries in image, (b) Largest boundary in image

1. **Fractal Analysis**

Now that the image has been processed and we have the boundary of the particle, we can begin the fractal analysis. The fractal analysis is conducted in the following manner. The initial stick length (blue line) is decided as the diameter of a circle with the same area as our particle.

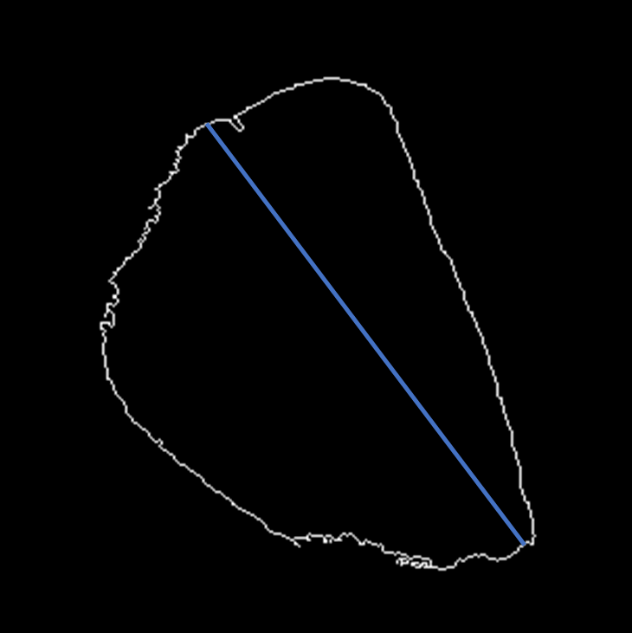


Fig 6: The initial stick length with the particle boundary

A point on the contour (Point A) is chosen as the starting point for the analysis. From this point A, we extend a line to the next point on the contour (Point B). If the line is longer than the stick length, then we add the length of the line to our perimeter and take point B as our new point A. We then repeat this process until we have gone through every point on the contour. Then store this as our perimeter.

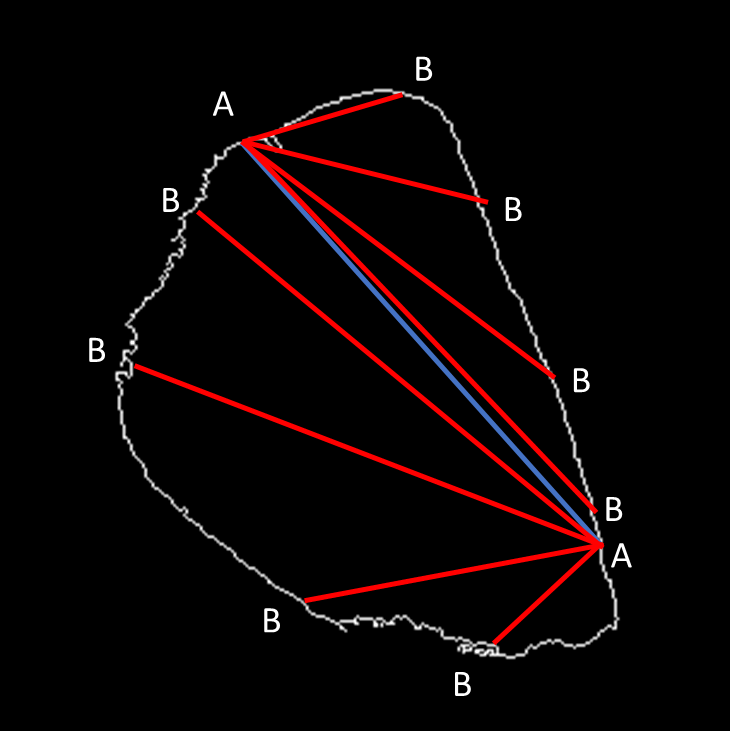


Fig 7: The repeated process to find the perimeter

This process is summarized in the below flowchart.

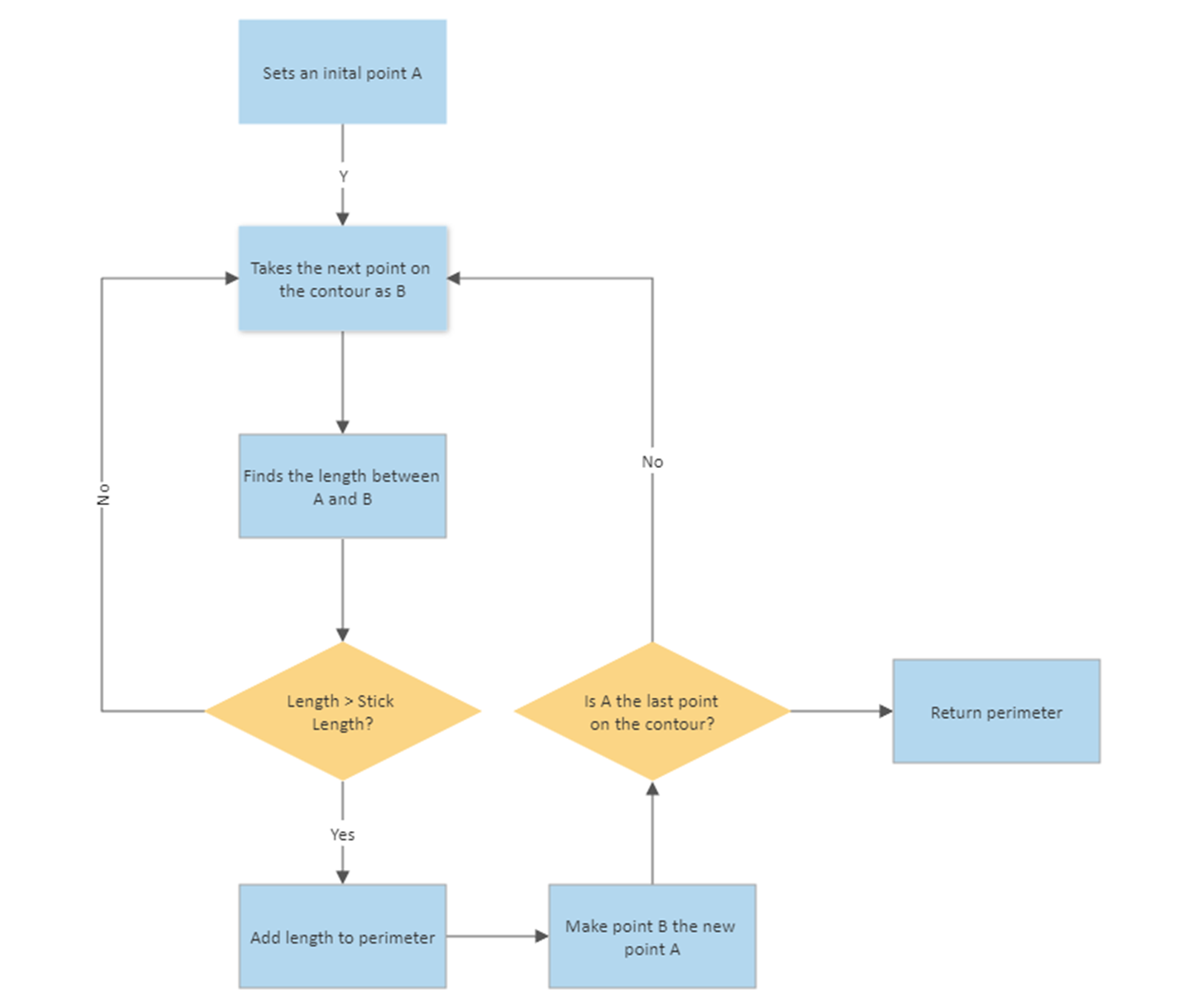


Fig 8: Flowchart showing the fractal analysis process

Once we find the perimeter for a specific stick length and a specific initial point A, we find the perimeters while taking every single point on the contour as our point A and finding the respective perimeter. We then take the smallest possible perimeter found for a specific stick length and we assume that to be our perimeter for that stick length.

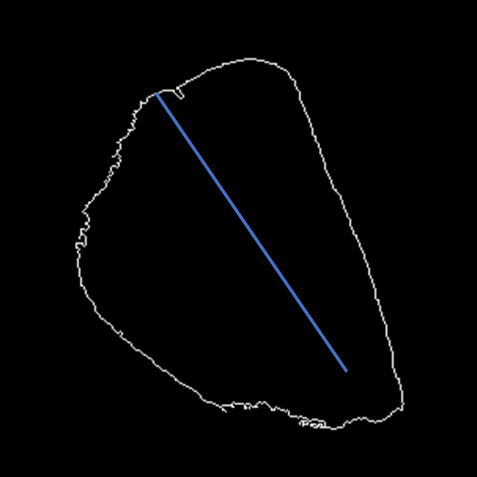


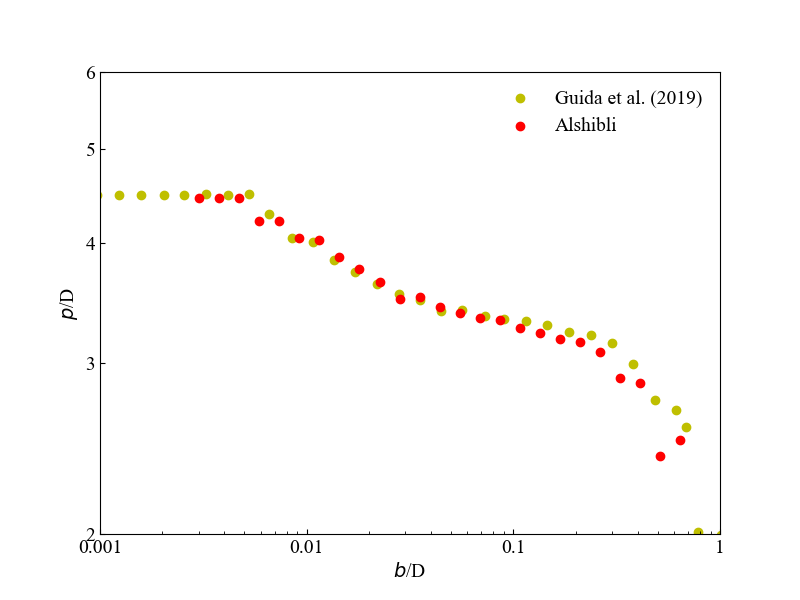
Fig 9: The shortened stick length with the particle boundary

Once we find the perimeter for a stick length, we then reduce the stick length and repeat the above process. We then track 2 ratios, the perimeter to initial stick length ratio and the current stick length to initial stick length ratio.

1. **Interpreting the data**

The next step is to create a scatterplot of all points on a log-log graph with the Current Stick length/Diameter as the X value and the Perimeter/Diameter as the Y value.

A graph of a function

Description automatically generated ****

(a) (b)

Fig 10: (a) Graph in the paper, (b) Graph from the code

After creating the scatterplot, we then select 4 points between which we can see an immediate shift in slope and create 2 linear regressions between those points. We then receive the R-Squared values for the 2 lines and are given the option to redraw the lines if unsuitable.

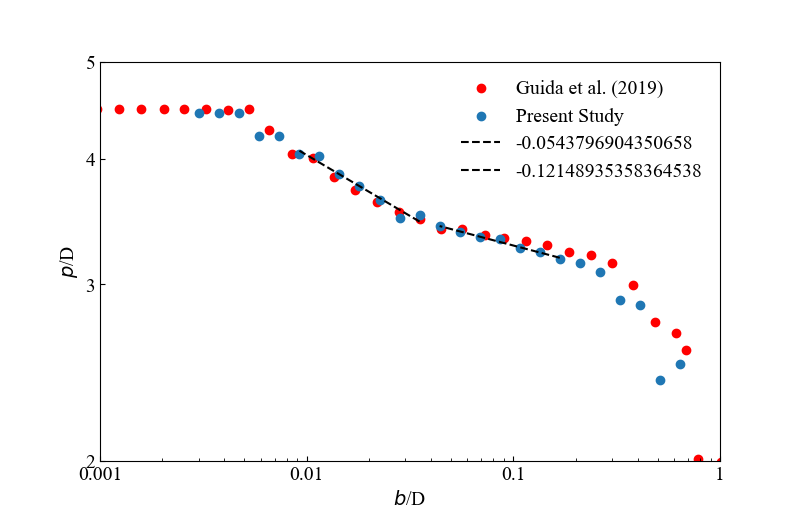


Fig 11: Graph with 2 linear regressions

From the graph, we say that the slope of the right line is an indicator of the angularity(m), and the slope of the left line is an indicator of its roughness(u).

To determine the overall form(M) of the particle, we compare it to a circle to get a quantitative value. The Y value at the point where the slope changes in the graph, is used to determine the overall form of the particle. We first take both the Y value that we get in a circle and the observed particle and subtract both by 2.

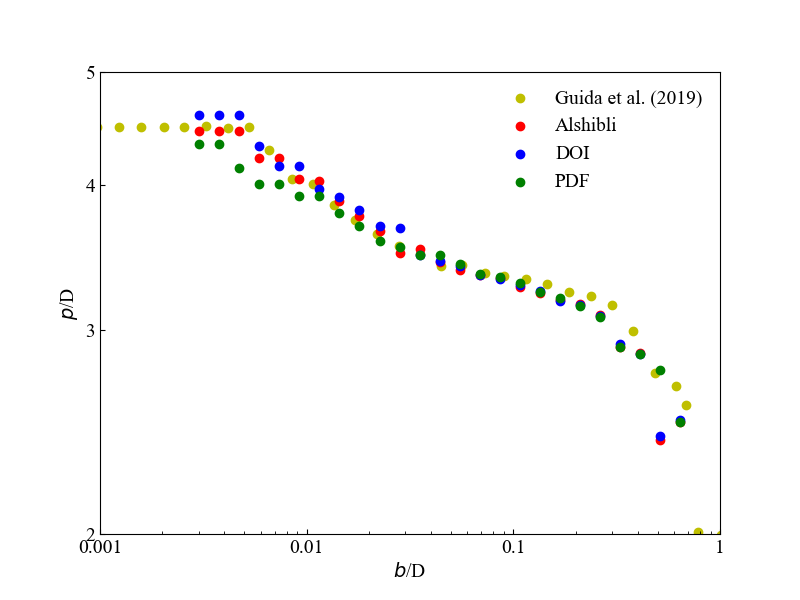
-(2)

We then divide the circle’s value by the particle’s value which gives us the overall form of the particle.

-(3)

**Results**

The first thing we look at is the influence of the resolution i.e., the number of pixels in the image, on the fractal analysis. Here we can see the yellow dots are the particle data that is obtained from digitizing graphs in the paper (Guida et al. 2019). The Alshibli image had 695 pixels in its boundary. The DOI image had 795 pixels in its boundary. The PDF image had 905 pixels in its boundary. Despite different resolutions and pixels, the values for all 3 are similar.



Alshibli

DOI

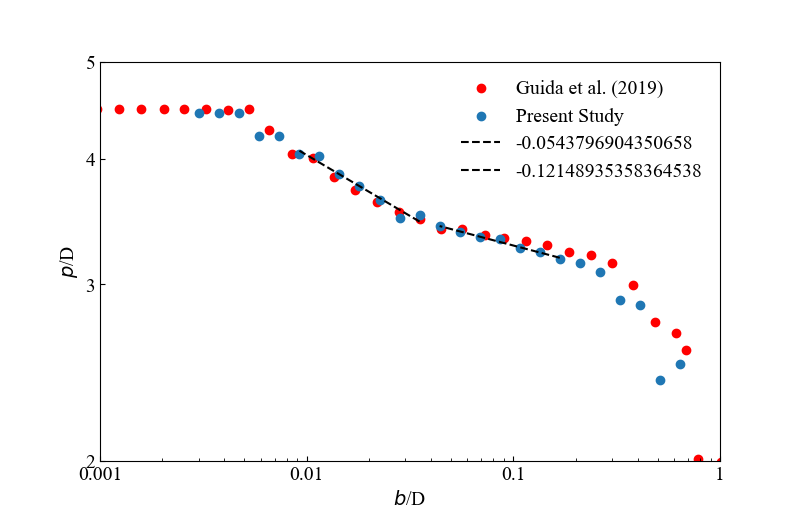


PDF



Fig 12: Comparison of the same picture in different resolutions

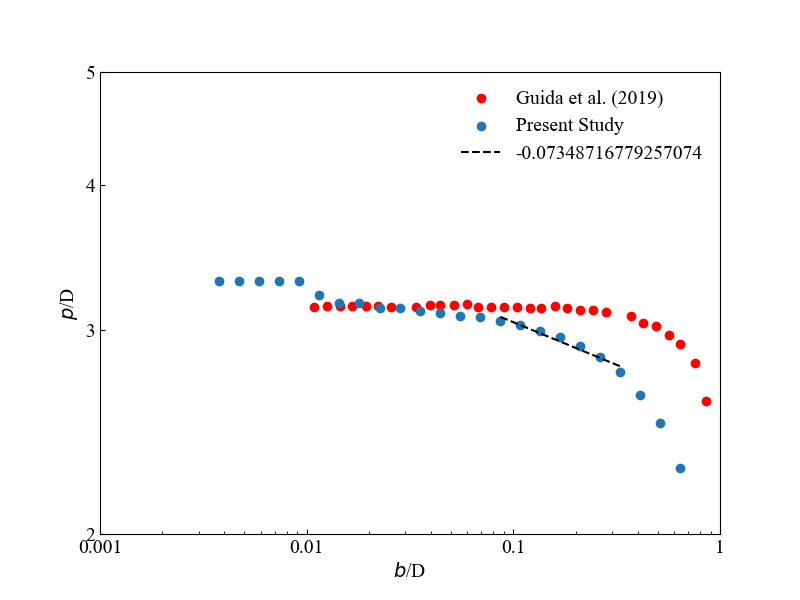
The following graphs are the results of the code.



**m = 0.05**

**u = 0.12**

**M = 0.75**

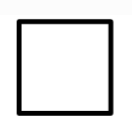
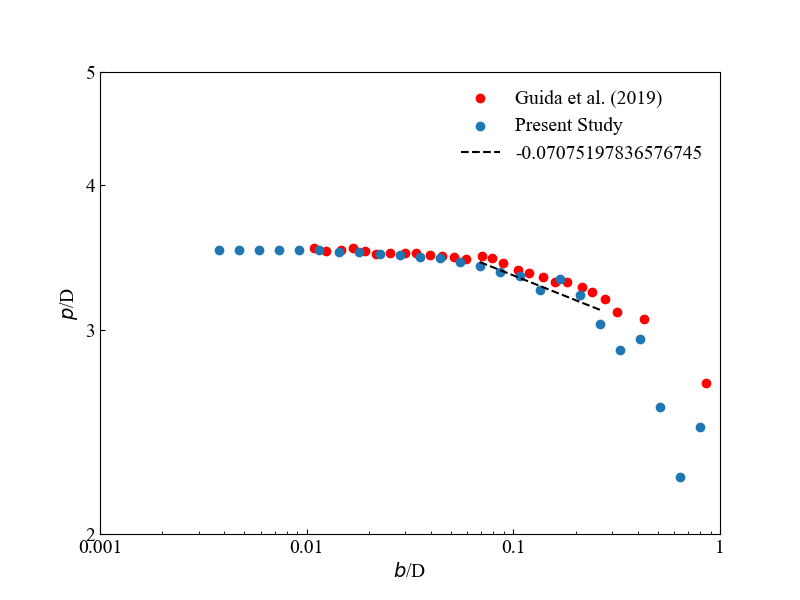
Fig 13: log(*p/D*) versus log(*b/D*) of Toyoura Example

**m = 0.07**

**M = 1.24**

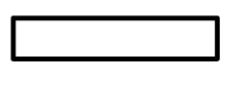


Fig 14: log(*p/D*) versus log(*b/D*) of Circle



**m = 0.07**

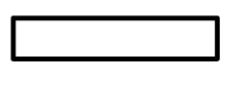
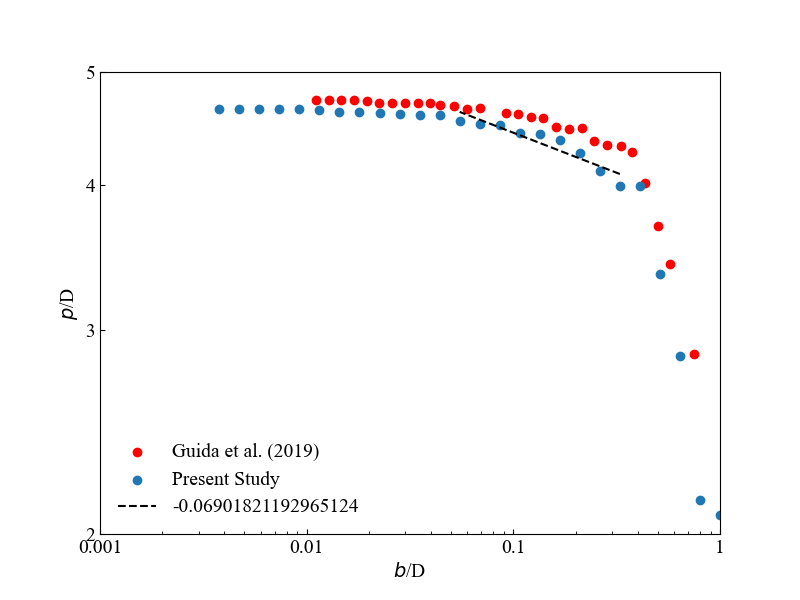
**M = 0.78**

****

**m = 0.07**

**M = 0.43**

Fig 16: log(*p/D*) versus log(*b/D*) of Rectangle

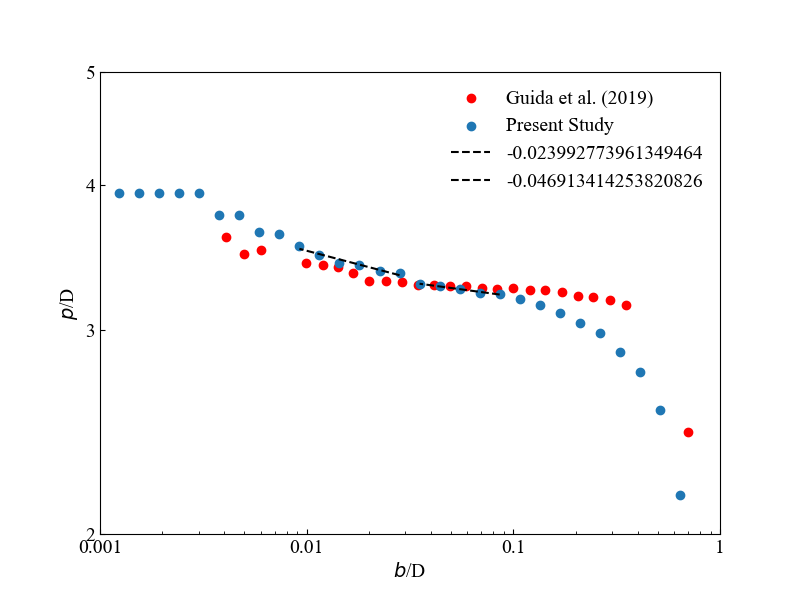


**m = 0.07**

**M = 0.43**

Fig 15: log(*p/D*) versus log(*b/D*) of Square

Fig 18: log(*p/D*) versus log(*b/D*) of ASTM 1



**m = 0.02**

**u = 0.05**

**M = 0.87**

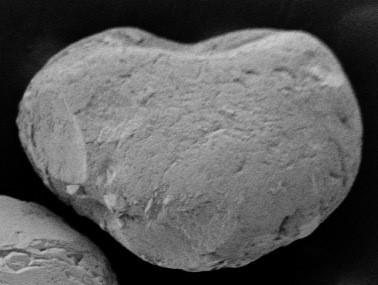
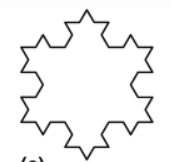
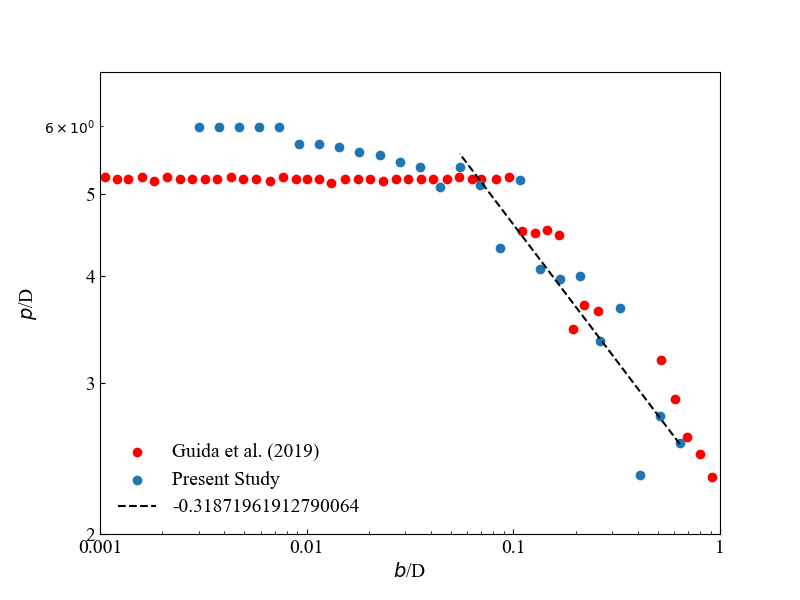


Fig 17: log(*p/D*) versus log(*b/D*) of Snowflake



**m = 0.32**

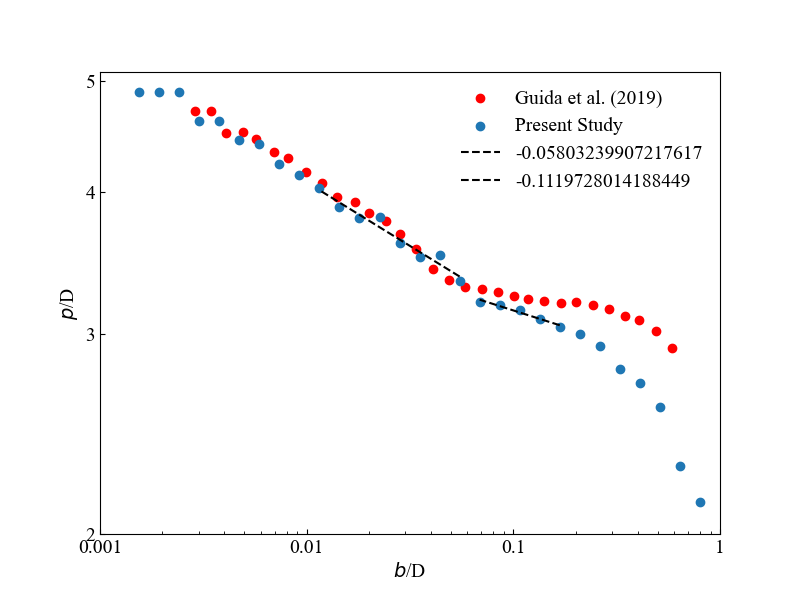
**M = 0.35**

6



Fig 19: log(*p/D*) versus log(*b/D*) of ASTM 2

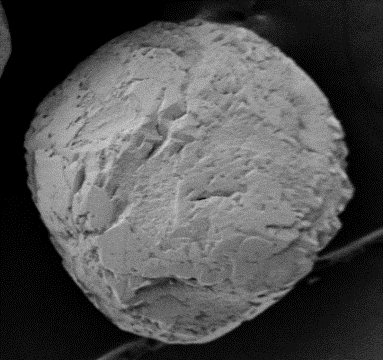
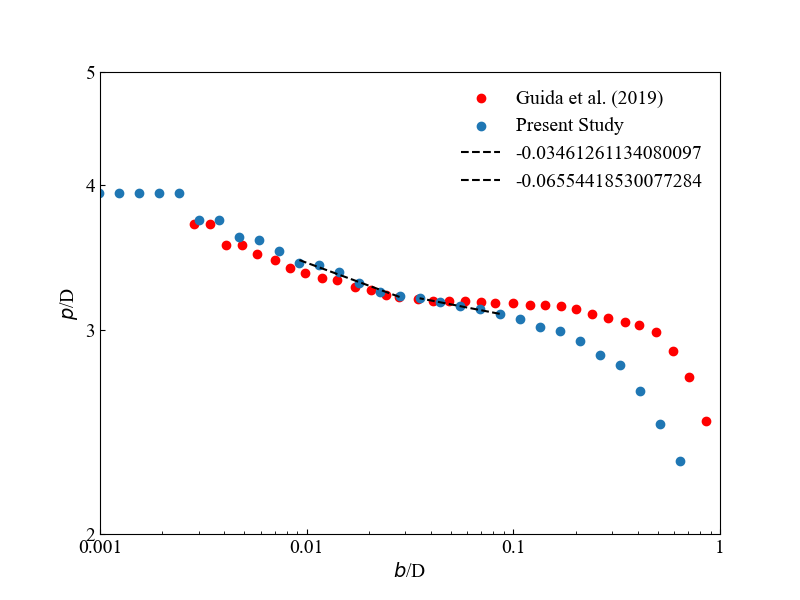
Fig 20: log(*p/D*) versus log(*b/D*) of ASTM 3



**m = 0.06**

**u = 0.12**

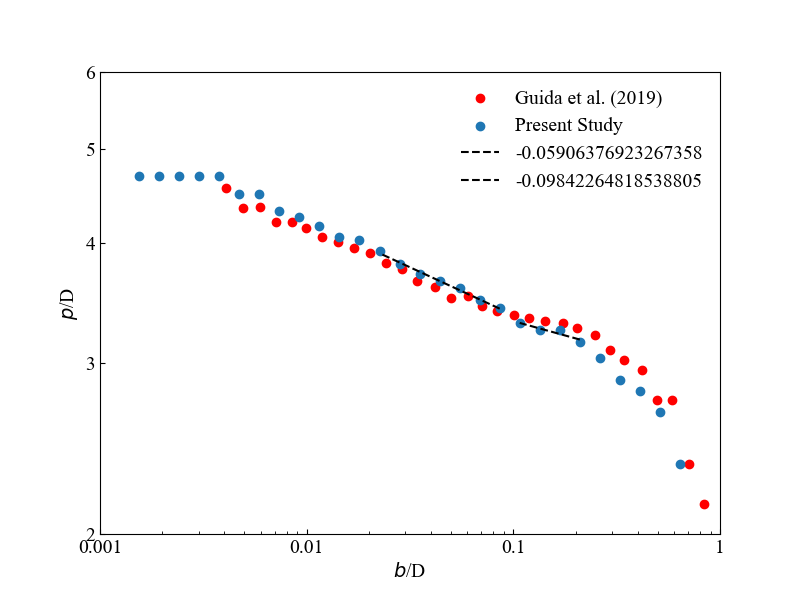
**M = 0.84**



**m = 0.04**

**u = 0.06**

**M = 0.96**



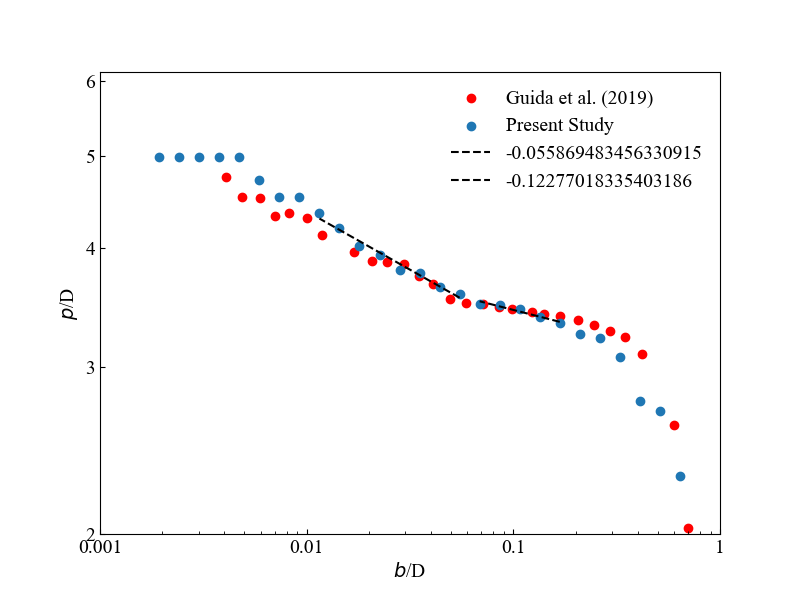
**m = 0.06**

**u = 0.10**

**M = 0.82**

Fig 22: log(*p/D*) versus log(*b/D*) of Toyoura 2

Fig 21: log(*p/D*) versus log(*b/D*) of Toyoura 1



**m = 0.05**

**u = 0.12**

**M = 0.72**



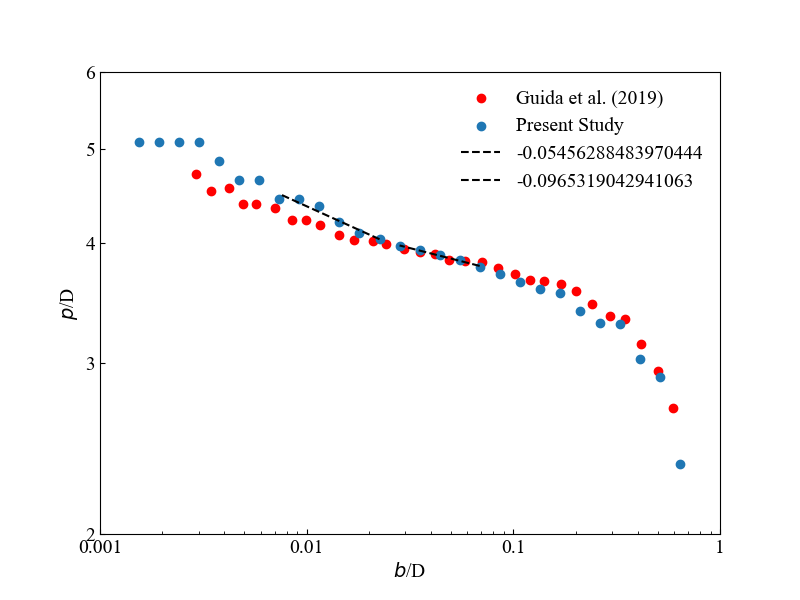


Table 2: Comparison between data from Guida et al. 2019 and data from the present study

Fig 23: log(*p/D*) versus log(*b/D*) of Toyoura 3

**m = 0.05**

**u = 0.10**

**M = 0.59**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Material** | **M (Guida et al. 2019)** | **M (Present study)** | | **m (Guida et al. 2019** | **m (Present Study)** | 𝛍 (**Guida et al. 2019)** | 𝛍 **(Present study)** |  |  |
| **Toyoura Example** | 0.75 | 0.75 | | 0.04 | 0.05 | 0.13 | 0.12 |  |  |
| **Circle** | 1 | | 1.24 | 0.07 | 0.07 |  |  |  |  |
| **Square** | - | 0.78 | | 0.07 | 0.07 |  |  |  |  |
| **Rectangle** | - | 0.43 | | 0.07 | 0.07 |  |  |  |  |
| **Snowflake** | - | 0.35 | | 0.26 | 0.32 |  |  |  |  |
| **ASTM 1** | 0.87 | 0.87 | | 0.01 | 0.02 | 0.04 | 0.05 |  |  |
| **ASTM 2** | 0.84 | 0.84 | | 0.03 | 0.06 | 0.12 | 0.12 |  |  |
| **ASTM 3** | 0.95 | 0.96 | | 0.02 | 0.04 | 0.06 | 0.06 |  |  |
| **Toyoura 1** | 0.76 | 0.72 | | 0.05 | 0.05 | 0.11 | 0.12 |  |  |
| **Toyoura 2** | 0.81 | 0.82 | | 0.05 | 0.06 | 0.09 | 0.10 |  |  |
| **Toyoura 3** | 0.59 | 0.59 | | 0.05 | 0.05 | 0.08 | 0.10 |  |  |

**Conclusion**

The particle shape characterization of multiple particles has been done using the fractal analysis method through programming in the python language. The characterization has been completed for 11 different particles. The data collected and interpreted matches the data from the source paper (Guida et al.2019).

**References**

1. Barrett PJ (1980) The shape of rock particles, a critical review. Sedimentology 27(3):291–303
2. Guida, G., Viggiani, G.M.B. & Casini, F. Multi-scale morphological descriptors from the fractal analysis of particle contour. Acta Geotech. 15, 1067–1080 (2020)
3. Finding boundaries - <https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html>
4. Thresholding - <https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html>
5. Log-log plot with linear regression - <https://stackoverflow.com/questions/32536226/log-log-plot-linear-regression>
6. Gamma Correction - <https://pyimagesearch.com/2015/10/05/opencv-gamma-correction/>
7. R-Squared Calculation - <https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html>