Image Recognition of Granular Configurations

By Brendan Lockhart*, Advisor Rena Zieve

University of Maryland, College Park

Abstract

The present research is focused on identifying the mechanism behind variability in the angle of stability of granular heaps. To study this, a cylinder containing steel balls in two different configurations is rotated slowly. As the heap passes the angle of stability, an image is taken to be analyzed. In this paper I describe the improvements made to the image recognition code to facilitate computational analysis of these images.

*blockhar@terpmail.umd.edu

Introduction

It is the hope of this research that analyzing the effect of grain structure could lead to a more complete understanding of granular dynamics. A granular material is a conglomeration of discrete macroscopic particles large enough to not be affected by temperature fluctuations.[?] In the everyday world, granular materials are exceedingly common; examples include sand, pills, beans, and corn grains.[?] Because they are so common in industry a more complete understanding of granular dynamics could be exceedingly beneficial in decreasing transportation costs (e.g. by packing granules more effectively) and evaluating geological instability to prevent events such as avalanches. Granular materials possess a variety of interesting and non-intuitive properties that necessitate a unique classification as a state of matter distinct from the three fundamental states of matter. For instance, a single grain of rice would certainly be classified as a solid, but a conglomeration of these rice grains possesses properties not traditionally associated with solids. For instance, if a container full of rice is tipped to one side, eventually the grains begin to flow like a liquid, and if the container is shaken violently, the grains begin to behave much like a gas. Some of the properties of granular materials are unlike any of the other states of matter. For instance, a hollow cylinder filled with grains will have a pressure at the bottom, as expected, but interestingly this pressure asymptotically approaches a maximum as further grains are added. The cause of this phenomenon is that the pressure from new granules becomes increasingly distributed along the walls of the cylinder.[?]

Another phenomenon present in granular heaps is that the angle of the heap at which the granules begin to flow is extremely variable.[?]

The exact causes for this variability are unknown at present. To study this phenomenon, previous researchers constructed a cylinder 14 inches in diameter filled with steel balls 1/8 inch in diameter. Some of these steel balls were welded into dimers consisting of two silver balls, while others were welded together to form hexes consisting of seven balls. These hexes were then painted green to facilitate image recognition.

Both faces of the cylindrical container were Plexiglass, facilitating observations of the grain configuration. This cylinder was sufficiently shallow that only a single layer of balls would fit. As this cylinder is rotated slowly, eventually the angle becomes large enough that the grains begin to flow. The angle at which this occurs is called the "angle of stability". Once the cylinder reaches this variable angle of stability a web cam is used to record an image of the grains immediately before and after the avalanche occurred. Unfortunately, the grain compositions are too complex and variable to quantitatively analyze any relevant patterns influencing the angle of stability using a single image. As a result, we need to analyze large numbers of these images computationally.

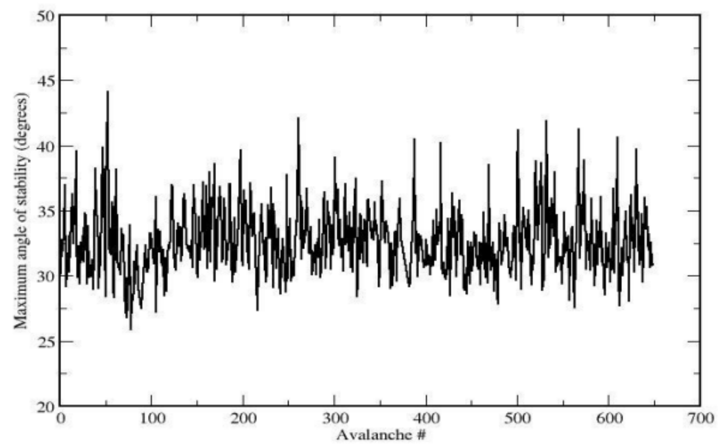

Figure 1: A graph of the angle at which the grains began to flow versus the avalanche number. Note the large variability of this angle in between successive measurements. This was done using single ball bearings, but the effect holds for the current apparatus.

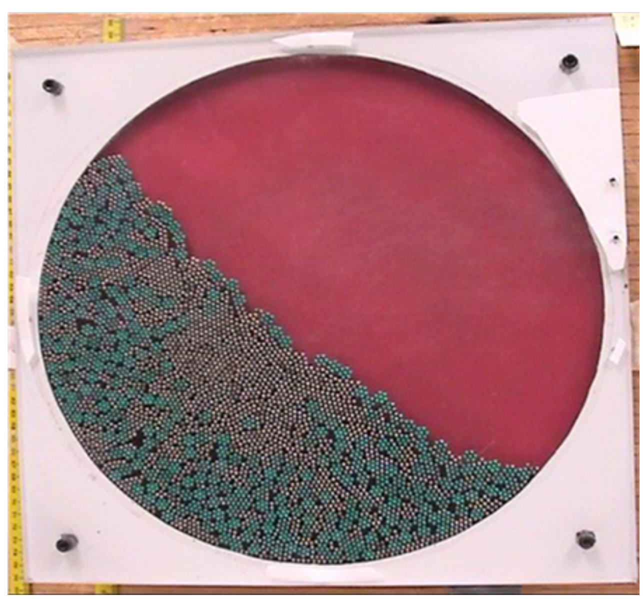

Figure 2: The experimental apparatus used in the present research. This cylinder contains both silver dimers and green hexagons.

To do this, previous researchers designed a code to computationally recognize where in the cylinder the different grain types were. Although this code initially performed well, it needed to be ported from the programming language "IDL" to "Python". In the process, the code accrued a variety of errors and was no longer functioning as designed. This summer I worked to identify errors within the code to improve the image recognition abilities of this program.

Method

The program involved numerous subroutines, many of which either remained functioning following the port to Python or had been debugged to work properly before the Summer of 2014. The general procedure of this code is to:

1. Find cylinder boundary

This stage operated by having multiple "rays" proceed radially outward from the center of the image (but not necessarily the center of the cylinder). Each ray would observe the strong white color of the cylinder's boundary, and recognize it as an edge. The distance of each ray path from the guess center to the white boundary was used to refine the estimated location of the cylinder's center.

2. Find the centers of the balls

This aspect of the code was rewritten to locate local maximums in one of the Red, Green, or Blue color spectrums of a certain width and above a certain threshold. The widths and threshold were manually chosen to best work with the data. As a result, there is the potential that a new filming environment will necessitate a new choice of parameters. Fortunately, this method remained effective on recordings taken on different dates, so a reevaluation of these parameters may not be necessary, unless a camera with a different resolution is used.

3. Classify the balls by color

The classification of ball color begins by identifying green balls. This is done using machine learning constructs called neural networks. Neural networks are a growing topic of research within computer science. They are not programmed in the traditional sense, but are rather "trained" to behave correctly.[?] This training procedure is done by inputting data into the neural net and using the difference between the neural network's result and the correct result to determine how much the neural network needs to be changed. This learning continues iteratively until the neural network behaves as desired.

The remaining ball centers are analyzed using a more conservative heuristic measure which tests if the color is above a certain threshold to determine if the ball is a silver ball. This method of finding silver balls is more conservative in order to allow balls that may be green to be labeled as "fuzzy". During the porting process, the identification procedure was incorrectly transcribed. As a result, the number of silver identifications was far below what it otherwise should have been (see Table 1). Upon appropriate repair the silver identification was improved, at the expense of no "fuzzy" ball types being identified. This is a serious issue as further steps in the code assume that the silver identification is accurate. As can be seen in Figure 4, the silver identification does at times identify balls incorrectly. To remedy this, the threshold for an identification of "silver" instead of "fuzzy" should be changed.

4. Identify hex centers

In this stage the program identifies which of the green balls are at the center of a hexagon. Five neural networks are used in this stage of the analysis, each corresponding to different data about the ball. The first three are 13x13 image samples taken from around the center in a R/G/B channel. The fourth uses a summation of every color channel in that 13x13 grid, and the fifth uses position information including the number of surrounding green balls, the standard deviation of the distance between the candidate hex center and neighboring balls, the number of neighbors, etc. The ball is labeled a hex center if all five neural nets agree that the ball is a hex center. If only four neural nets find the ball to be a hex center, then the ball is labeled as a candidate for further inspection.

Results

After the program had been ported to Python, the results were unsatisfactory for granular configuration analysis. As can be seen in Figure 3, many of the balls remain undetected, and many of the centers are not found to be on the true center, but are offset slightly.

The initial stage of the analysis, where the program identified the border of the cylinder, was functioning optimally and did not require any adjustment. The second stage, which involved locating balls within the image, was not operating sufficiently well and required improvements. I rewrote this program to identify balls by applying a multidimensional scipy filter to the image in each of the color spectrums. This found local maximums of a given diameter in the two dimensional array. If these local maximums are above a set threshold and within the bounds of the cylinder, they are then added to the list of balls. As can be seen by comparing Figures 3 and 4, this process recognizes far more balls than the previous method, especially near the perimeter of the cylinder. Unfortunately there are some false positives, particularly on the periphery of another ball.

Following a recommendation from the researcher who

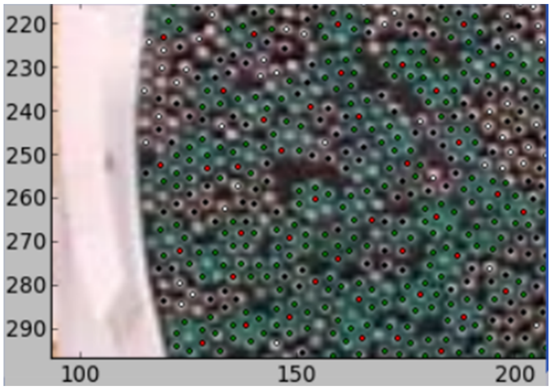

Figure 3: A sample of the image recognition program's results after being converted to IDL from Python and before improvements undertaken in the Summer of 2014. Here, the dots superimposed onto the image correspond to the software identifying which class of ball the program has identified. Here, white, green, red, and black correspond to silver, green, hexcenters, and "fuzzy" balls, respectively.

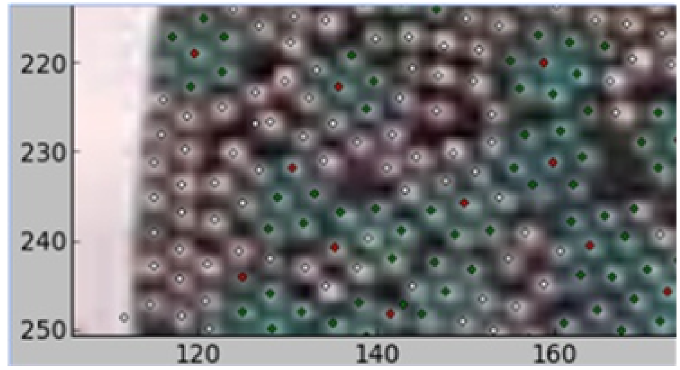

Figure 4: A sample of the image recognition program's results after this summer's updates. The code now recognizes far more balls, but does have a few false positives located on the periphery of true balls. The code no longer uses the fuzzy classification.

	Previous Analysis	Current Analysis	Actual
Balls found	4084	4591	4383
Green balls found	1879	2160	2191
Silver balls found	839	2431	2192
Fuzzy balls found	1374	0	
Hexes found	310	338	313
% True centers found	67.4%	82.9%	

Table 1: A comparison of the image recognition's ability to recognize various types of balls. This table suggests that the program is now far more effective.

previously worked on this project, I worked on identifying an error within the silver identification step. Although I did find such an error, fixing the error came at the expense of identifying fuzzies. In effect, fixing the error altered the value of the threshold needed to classify a ball as a silver. This is a serious error, as future stages within the program assume that silver balls are definitely silver. This threshold should be improved in future revisions of the code.

Although a qualitative assessment of the improvement proved amiable, I wanted a more quantitative assessment. The fact that many hex centers found by the program are false positives and an approximately equal number go unidentified suggests that simply comparing the total number of hex centers found to the correct number may lead to misleading results. To remedy this, I designed a code to compare programmatically obtained findings to manually inputted results. A comparison of the findings is shown in table 1.

As you can see, the efficacy at identifying hex centers has increased markedly. Although it isn't at a high enough rate to be considering fully functioning, it marks a substantial improvement.

Future Work

Looking ahead, the silver identification program really needs to have its threshold for identification either lowered to be made more conservative or made more sophisticated. This is because fuzzies are an important aspect of the analysis, and as can be seen in Figure 4, some green balls are incorrectly labeled as silver.

Secondly, I would like to see the neural networks retrained to best work with the current method of locating the balls, perhaps using a different network architecture. I had created one program to train a new neural network using the manually identified hexes, but I was unable to get it fully functioning.

Thirdly, I believe that the software could be further improved by adding a heuristic to remove false-positive balls. These happen characteristically at the periphery of true ball centers and at the joining point in between two balls. I had attempted to filter these out by replacing very close ball pairs with a single ball between the two, but was met with unsatisfactory success. Perhaps a better method for removing these excessive identifications would consist of observing the overall color gradient in the region.

Acknowledgments

I would like to thank my advisor Rena Zieve, Zuni, and the National Science Foundation

References

[1] http://london.ucdavis.edu/ reu/reu08/kelly.pdf.

[2] Sidney R. Nagel, Heinrich M. Jaeger, and Robert P. Behringer, "The physics of granular materials," Physics Today 49:4, 32 (1996).

[3] Dvornik J. and Lazarevic D. "Loading of granular material on silo walls," in *Proceedings of the 1. Conference on Applied Mathematics and Computation* ed. M. Rogina, V. Hari, N. Limi and Z. Tutek (Dubrovnik, 1999), p. 215.

[4] Olson, J. Priester, M. Luo, J. Chopra, S. and Zieve R., "Packing fractions and maximum angles of stability of granular materials," Physics Review E. 72:3, 31302 (2005).

[5] S. Rowley, H.A. Baluja and T. Kanade. "Neural network-based face detection," IEEE Transactions on Pattern Analysis and Machine Intelligence 20:1, 23 (1998).