Image Analysis of Granular Piles

Daniel A. Silva*
California State University Long Beach

The highly complex and disordered structure of granular piles have prevented a comprehensive model to explain all of their properties such as segregation of a binary mixture or the angle at which an avalanche will occur. In order to investigate the properties of granular piles more effectively we need to be able to recognize patterns in the avalanches. Imaging software is used to find hexagons and doubles in the granular piles.

I. INTRODUCTION

Granular piles have some very interesting properties such as segregation and the angle of stability of a granular pile. There have been experiments that have investigated these properties. However the chaotic nature of granular mixtures has prevented a model that predicts general behavior based on the configuration of the individual particles. In order to develop a model which can explain these properties we need to be able to collect data on the positions of the individual particles of each granular mixture. In order to improve measurement statistics we need to study a large number of avalanches. Collecting large numbers of avalanches becomes impractical unless the analysis is highly automated.

In this experiment we place a binary granular mixture of 1/8 " ball bearings in a quasi two dimensional system. This is created by confining the ball bearings in between two sheets of Plexiglass. The mixtures consisted of doubles, two silver ball bearings which were welded together, and hexagons,seven green ball bearings that were welded together. The bearings were painted green in order to better recognize them using the imaging software. The plexiglass container is rotated by a drum at 500 μ Hz. The frames immediately before and after each avalanche was saved and stored for analysis.[1]

The images were analyzed by an imaging software that employed neural networks and heuristics. The software was written originally written in IDL and designed to find the individual ball bearings, doubles, and hexagons. The code was then translated into python. This code combines two different techniques to analyze the granular heaps. Then it combines the results of each.

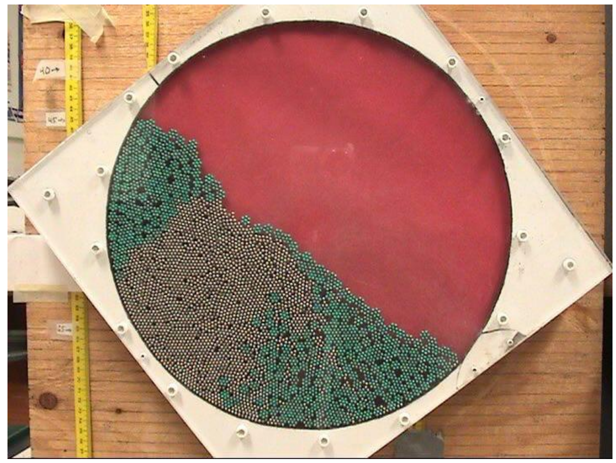

FIG. 1. Picture of binary mixture of hexagons and doubles before rotation in a drum.

II. IMAGING SOFTWARE

The imaging software for finding the hexagons is a multistage process. The imaging software also uses a combination of two methods and combines the results of each. The two techniques used are neural networks and a heuristic scoring algorithm. The combination of these two techniques improves the accuracy of the imagining software. The imaging software runs in stages. The first stage is the setup. Then we have a neural network stage, and then a scoring algorithm.

A. Setup

We have defined a class whose attributes contain all the information that we need for an individual ball bearing, such as its neighbors, whether it is the center of a hexagon and other important information which will be discussed. The first step is finding the centers of each individual ball bearing. The brightest point of each ball bearing is labelled the center. The center of each ball is stored as coordinates on a plane. We next form an irregular lattice where each ball is a node in a Delaunay triangulation. In this way each ball is connected to its nearest neighbors. Each ball's nearest neighbors are stored in an adjacency matrix.

We next find the color of each ball through a threshold and using neural networks. First we find all the green balls using a neural network. We then find all the silver balls using a conservative threshold. Because of the conservative threshold some silver balls are missed. Some green balls are missed as well. Balls are labeled fuzzy if their color is not determined. Green balls that have two

* 2013 Research Experience for Undergraduates,
University of California, Davis

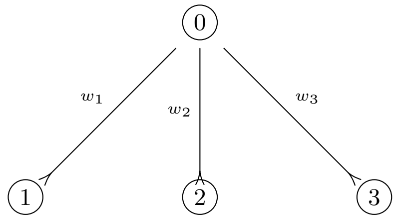

FIG. 2. Neuron with weights denoted by w and three inputs.

or more silver balls as neighbors are stored as edge balls, that is possible edge balls of a hexagon. Green balls that have no silver balls are stored as possible hexagon centers. Once we have all this information we then go on to the neural network stage.

B. Neural Network Stage

Finding all the ball bearings in the heap turns out to be non trivial. Different conditions of each experiment can affect the images. For instance the lighting of the room, the distance of the camera from the tumbler, and using a different camera. These all will affect the images. One way to deal with these complications is to use neural networks. Neural networks are computational models of our biological brain. They are particularly suited for image analysis in which there is a lot of noise. Neural Networks are directed graphs. The simplest neural network is the feed forward network. This is the type that we use. A feed forward network has links only directed forward from input nodes to output nodes. Neural Networks have the ability to learn from data; this is what makes them so well suited for image analysis. Training a neural network is needed to be able to use them. The neural network gets information from the data and it will change the weights of each node. The information of the neural network is stored in these weights. Referring to fig. 2 we can see that the input to each node is described by the sum of the weights of each node times the the output of each node. The output of a given node is a function of the input. So

$$S_j = \sum_i w_{ij} a_i \tag{1}$$

$$a_j = f(S_i) \tag{2}$$

The neural networks were trained by bitmaps of hexagons as well as non-hexagons.The neural networks were trained by Python FFNet module. The Fortran code that was generated was originally translated into IDL but then it was translated into python. The neural networks created formed a committee. Each ball in an image is run through this committee of five neural networks. Each network will return either a 1 or 0. It will return a 1 if the network decided it was a hex center and 0 if it didn't. If 5 out of 5 of the networks return a 1 then the ball is labeled a hex center. If 4 out of 5 of the networks return a 1 then it is labeled a candidate.

C. Heuristic Scoring Algorithm

After the neural network stage the scoring algorithm stage occurs. The scoring algorithm is rather complex so it is worth going over in detail. First we make a list of orphan balls. Orphan balls are balls that are fuzzy or green and do not belong to a hexagon. A list of balls that are edge balls is made. Once we have this we proceed with the scoring algorithm.

1. Assign Scores to Orphans

Orphans are given scores depending on whether they are green or fuzzy. If the ball is an edge ball it is increased by a certain multiplier. The neighbors are given scores depending on ball's score divided by the product of the distances of each neighbor and number of neighbors it has.

2. Tweaking Candidate Score

The score of each candidate is changed by dividing the score by the maximum distance of one its neighbors exponentiated to some power. This will exclude candidates with neighbors that are far away, since hex-centers should not be very distant from other green balls.

3. Selection

The list of candidates is iterated through and compared locally. If the ratio between a ball's scores and its nearest neighbors reaches a certain cutoff it is added to the list of hex centers. This is used over just a threshold score because in regions where there is close packing of hexagons the scores of its neighbors will be high while in a regions with sparse amount of hexagons its neighbors scores will be low.

4. Cleanup

Hexcenters that are too close to each other are both deleted from the list. As we find hexagons the amount of positions where hexagons can be placed decreases. Hexagons were missed in this stage because in regions with densely packed hexagons a small error could affect it going ahead. If we mislabeled a hex center then in

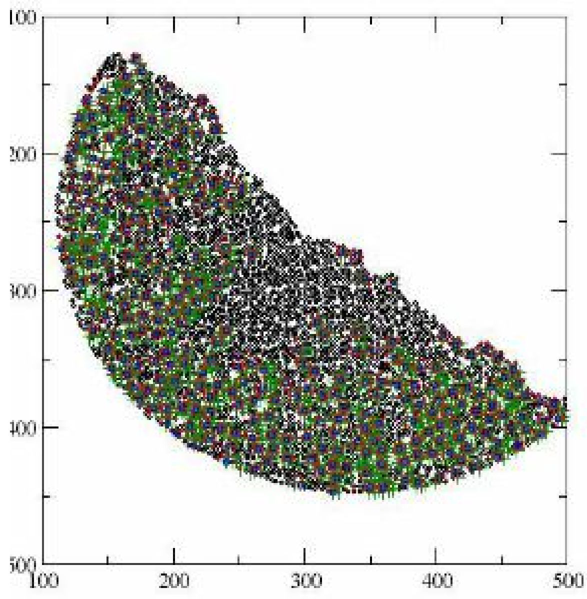

FIG. 3. Plotted output of the imaging software

the locally surrounding regions there would be no legal place to put a hexagon.

In order to deal with this error neural networks are used. Neural networks will find more hex centers than are actually there so it is coupled with the scoring algorithm for best results. The three stages of the imaging software is the setup, neural networks coupled with the scoring algorithm, and then just the scoring algorithm.

III. DATA ANALYSIS AND RESULTS

Granular piles have some very interesting properties such as segregation and the angle of stability of a granular pile. However analyzing thousands of pictures does not seem practical so this justifies the need to automate the analysis of the images. Originally the code was written in IDL but then needed to be translated into python. The code was partly translated. It was fragmented, it did not work together and some parts of the code were repeated in other sections. Scripts were written that would run through some parts of the code and change various things from IDL into python. A major part of translating was connecting the various parts of the program together. The beginning of this was started but not completed. After the program was connected together and it was running again the parameters needed to be changed a little, such as the threshold that would find the most amount of silver balls and find no green balls. The code now is running at 92% accuracy.

Granular Piles	Program	Actual
Number of Balls	4034	4351
Number of Green Balls	1693	2163
Number of Silver Balls	72	2188
Number of Fuzzy Balls	2269	

	Neural Network Stage	Scoring Algorithm	Total	Actual
Number of Hexagons	249	37	286	309
Accuracy	92 %			

IV. FURTHER RESEARCH AND CONCLUSION

Now the code is now working together as one program and produces fairly accurate results. There are several different avenues of further research and improvement to the current program.
The most obvious is to continue to improve the accuracy of the program. When it was originally written in IDL the accuracy was 96%. The discrepancy can be due to many factors. One of these might be that the threshold for finding silver balls needs to changed. The amount of silver balls it is currently finding is very low. To fix this should be straightforward. One just needs to find the right ratio of color that a silver ball has in terms of red, blue, and green.
Another important, maybe the most important, avenue of further research is training the neural networks again. Currently the neural networks are trained to analyze images from the an old experimental setup. Retraining the neural networks shouldn't be harder than using the python module FFNet on new images. The main justification for automating the analyzing of the images is to be able to analyze thousands of images and thus get better statistics. Also one of the main reasons for using neural networks is so one can bypass the changing experimental conditions. So it is important to be able to use this program to analyze the current experimental setup.
Another extension of this experiment would be to try different shapes other than hexagons. Since the method of neural networks and the scoring algorithm has been shown to work for hexagons it should also work for other shapes.

V. ACKNOWLEDGEMENTS

I would like to thank my advisor Rena Zieve. I would also like to to thank the National Science Foundation.

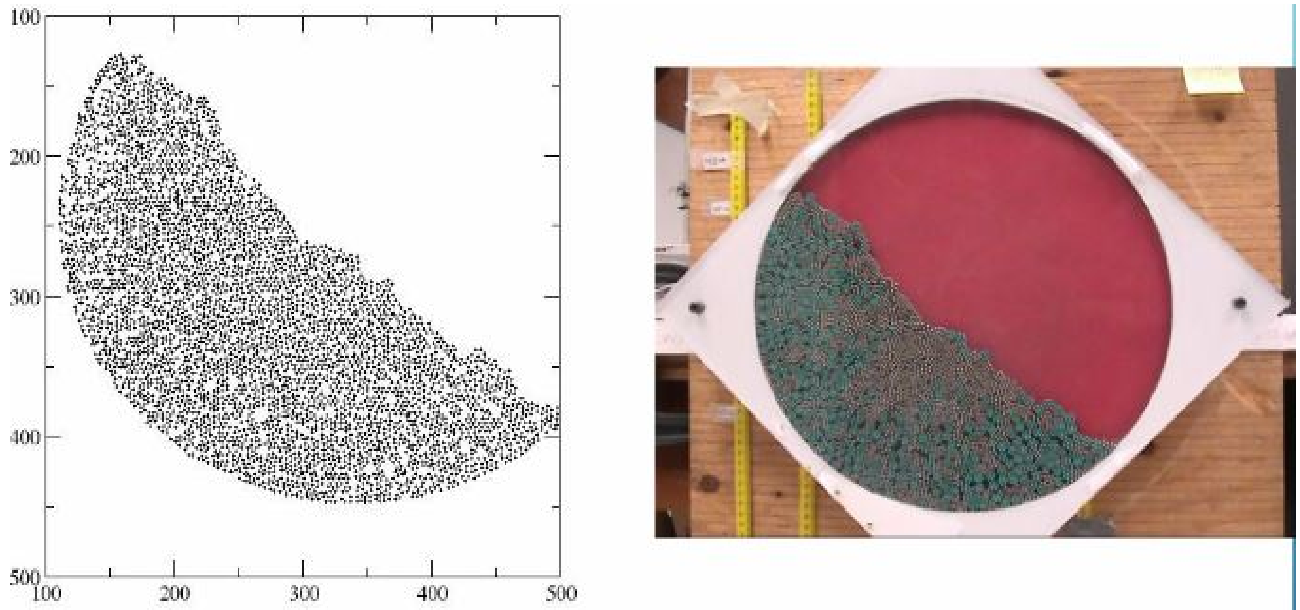

FIG. 4. Centers of Imaging Software

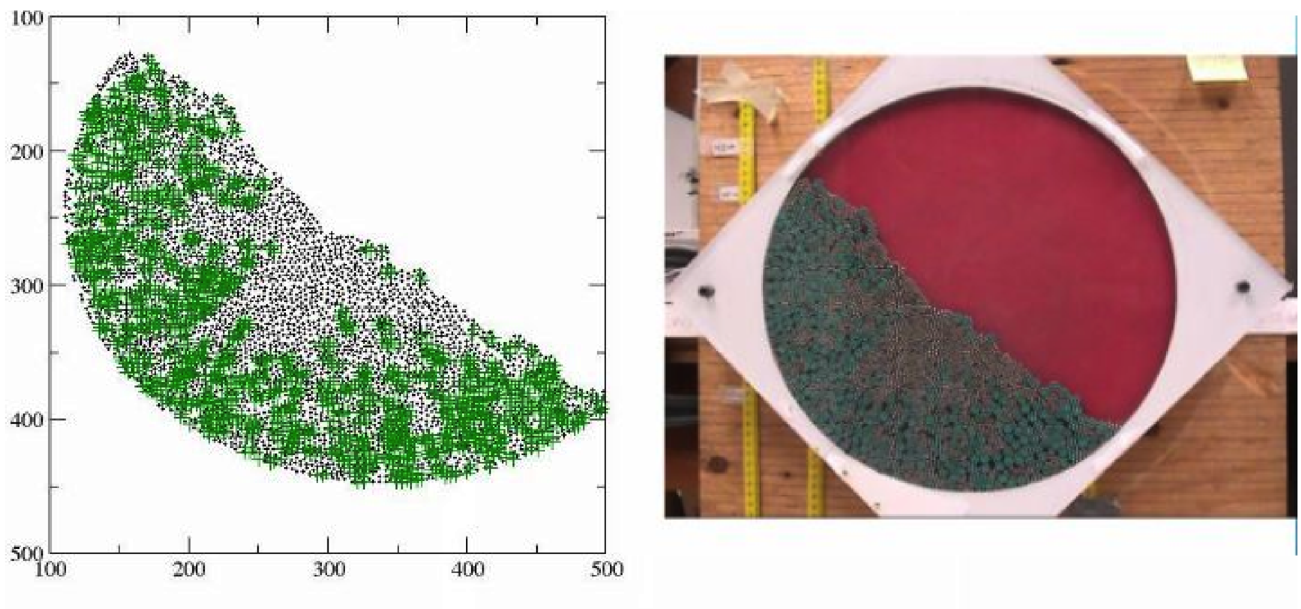

FIG. 5. Centers and Green Balls of Imaging Software

[1] Swartz et al. *Segregation and Stability of Granular Mixtures.*

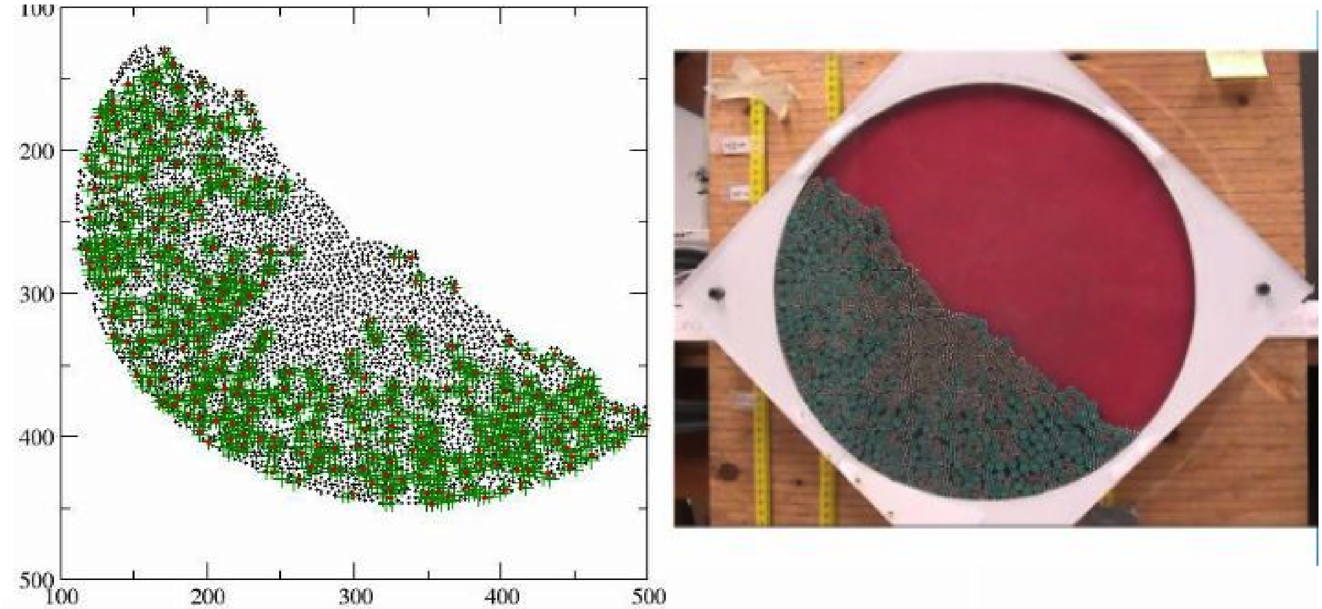

FIG. 6. Centers, Green Balls, and Hex Centers of Imaging Software