Image Analysis of Granular Mixtures:
Using Neural Networks Aided by Heuristics

Justin Eldridge
The Ohio State University[*]

In order to gain a deeper understanding of how individual grain configurations affect the general behavior of a mixture, the exact position of each grain must be recorded. Neural networks and heuristics are used to analyze images of granular mixtures, determining the location of each hexagonal grain. The results are visually represented by heat maps in order to give clues as to what trends should be investigated in future analysis.

I. INTRODUCTION

While granular materials have long been studied, a solid understanding of their complex behavior has yet to be formed. General trends, such as segregation and pile stability, have been observed and analyzed, but this sheds little light on how the specific configuration of individual grains affects the behavior of the pile as a whole. In order to develop a model that predicts general behavior given a specific configuration of grains, detailed analysis of the mixture involving the location of individual particles must be performed.

In this experiment, 1/8" diameter steel ball bearings took the place of grains. Seven green bearings, colored to aid in later analysis, were welded together to form hexagons, and two silver bearings were welded to make doubles. Various concentrations of hexagons and doubles were added to a drum. The bearings were confined to a single plane by two sheets of Plexiglass. The drum was then rotated at 500 μHz, and avalanches were recorded using a video camera. The frames immediately before and after each avalanche were then extracted from the video, and stored as image files (see Figure 1) [1].

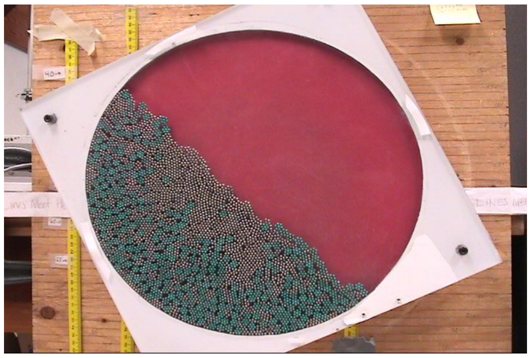

FIG. 1: An extracted frame, showing the drum containing half hexagons (green) by weight, half doubles (silver).

The images were then analyzed using an IDL program which found the angle of avalanche and density of the pile. A correlation was found between the concentration of hexes in the pile and the angle of avalanche, and a general pattern of segregation was observed, as the doubles tended to rise towards the middle of the surface. To analyze the mixture in detail, however, it was necessary to record the location of each hexagon within the pile. This required new IDL code, and a combination of techniques.

II. TECHNICAL BACKGROUND

Autonomously finding the hexes was nontrivial. The most obvious challenge was posed by the tendency of hexagons to pack together. When this occurs, the border between two hexagons becomes ambiguous, and it is difficult to determine how many grains there are and where their centers are located, even by eye. In especially dense regions, common in mixtures with high concentrations of hexagons, a dozen or so hexagons might pack tightly together. This forces someone inspecting the mixture to find the hexagons on the outside of the clump first, and then work his way inwards.

The situation was further complicated by the quality of the images. Recording hours of avalanche footage in high definition was not technologically feasible at the time of the experiment, so each image was taken at a resolution of 640×480 pixels. As a result, each 1/8" diameter ball had a radius of two pixels. Because the steel balls reflected light, some silver balls located near the green hexagons would themselves appear green. Combined with the low resolution of the image, this resulted in patches of fuzzy information, where the color of individual balls was ambiguous.

Furthermore, there were subtle changes in the apparatus that made image analysis more difficult. The distance between the camcorder and the drum was not kept constant over the entire experiment, therefore the pixel radius of a ball is not the same in every data set. Differences in lighting also caused the brightness and saturation of images to fluctuate.

Finally, the pool of potential approaches was limited by the amount of data to be processed. While there was no strict limitation on running time, there were questions as to how long a stochastic method, such as Monte Carlo, might take to converge, given that there were roughly 4000 balls per image and 3000 images in total. Ultimately, a Monte Carlo method was avoided because no proper heuristic for error minimization could be found, largely due to the fact that the image data being analyzed was less than ideal.

The naïve method of finding hexes would be to establish some heuristic involving the angle between neighboring balls, since the centers of the balls welded into the hexagons are al-

[*]2009 Research Experience for Undergraduates, University of California, Davis

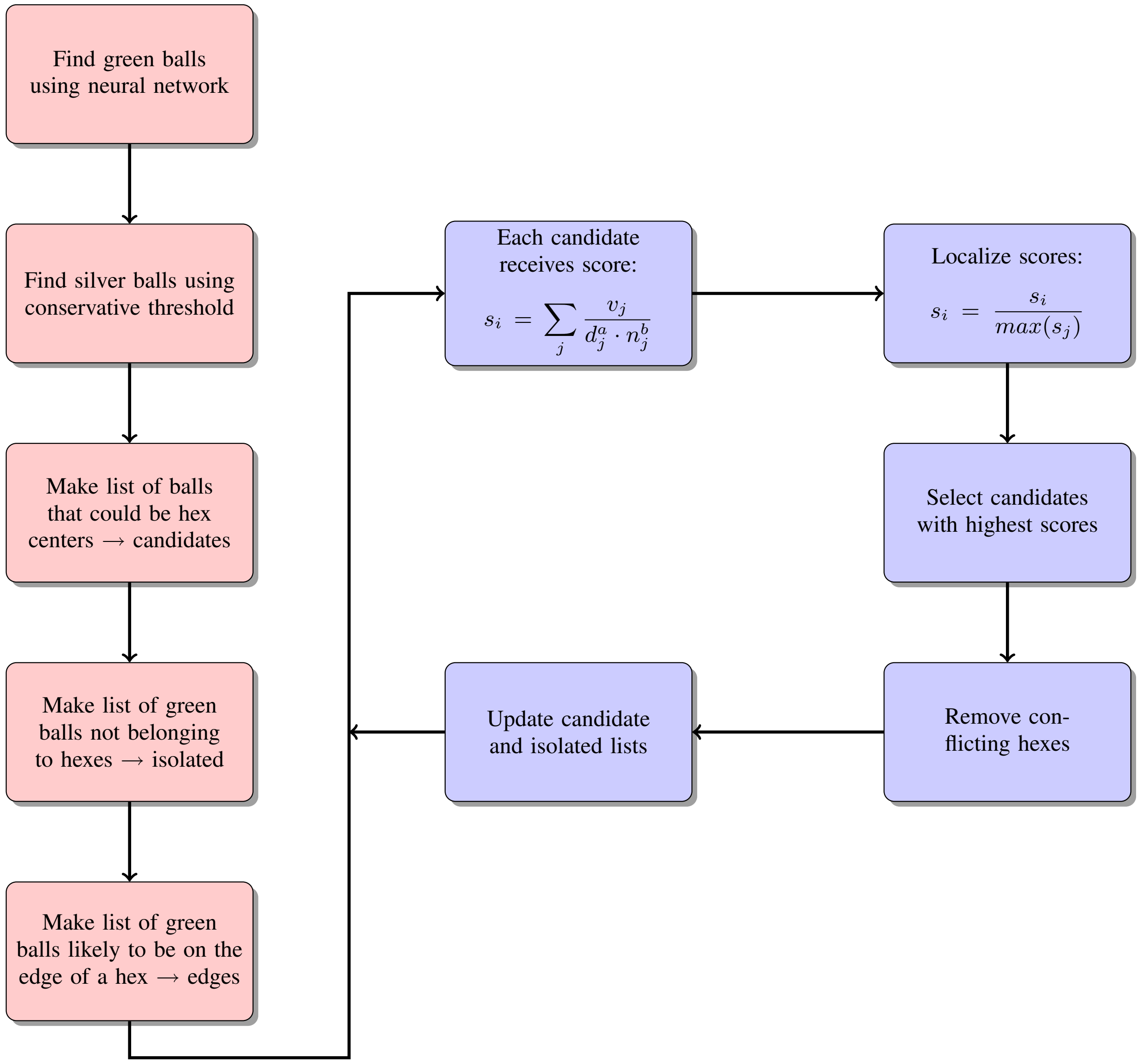

FIG. 2: A summary of the scoring algorithm. The scoring loop (in blue) is repeated twenty to thirty times. After each run, the list of candidates and isolated balls is narrowed, in effect "clearing up" the data for the next round.

ways separated by 60°, and are four pixels from the hexagon's center. Unfortunately, this is an idealization. Firstly, the algorithm previously developed to find the center of each ball works by finding the bright glare from the overhead lights. This is rarely located at the exact center of the ball, and, due to differences in lighting, is not constantly placed. Therefore, it is not possible to find the exact center of each ball with great accuracy, and any measurement involving distances or angles will incur a large margin of error. Secondly, even if this information was close to ideal, it does not work well to differentiate hexagons that are packed together.

Instead, a more complex set of heuristics is employed. A scoring algorithm was developed that emphasized the scores of obviously placed hexagons, such as those located on the outer edge of packed clumps. Before initializing the scoring loop, however, various categories are established to organize the data. First, each ball is found and placed into an irregular lattice. This is done by interpreting each ball in the image as a node in a Delaunay Triangulation. By doing this, each ball is connected to its nearest neighbors in a graph, and an adjacency matrix is produced.

Next, the balls are categorized according to color. Green balls that neighbor two or more silver balls are thought to be on the outer edge of a hexagon, and are added to a list of edge balls. Green balls that have no silver neighbors are added to a list of candidates; these are likely to be the center of a hexagon. Green balls that have no neighboring balls identified as hexagon centers are added to a list of isolated balls.

Once this information has been organized, the scoring algorithm proceeds as follows:

1. *Assign Scores*: Each green ball that has not yet been included in a hexagon is given a value. This value is increased by a multiplier if it is thought to be an edge ball. Each candidate neighbor is

then given a score that is directly proportional to the isolated ball's value, but indirectly proportional to the product of the distance between it and the isolated ball and the number of candidate neighbors the isolated has. Expressed in pseudocode:

for $ball \in isolateds$ **do**

$value \leftarrow 1$

if $ball \in edge$ **then**
$value \leftarrow value \cdot multiplier$
end if

;this will return the number of
;neighbors the ball has that are
;in the list of candidates
$n_c \leftarrow numcand(ball)$

for each $neighbor$ to $ball$ **do**
$d \leftarrow dist(ball, neighbor)$
$score \leftarrow score + \frac{value}{d \cdot n_c}$
end for

end for

2. *Selection*: If the ratio of a candidate's score to the maximum score of one of its neighbors reaches a threshold, it is added to a list of hexagon centers. The ratio is used over the flat score to account for the fact that regions dense with hexagons will naturally have high scores due to the number of surrounding isolated contributors, while solitary hexagons have low scores.

3. *Cleanup*: Hexagons that are found too close to one another are deleted from the list. Following this, the candidate and isolated lists are updated to reflect changes in the list of hexagon centers. By doing this, the number of potential placements for hexagons decreases as more are found, clearing up regions that were originally ambiguous.

This loop repeats a set number of times, and the results are analyzed.

This algorithm emphasizes finding hexagons that are obviously placed on the outside of clusters, and then works its way inwards, much like a human would do. It does this by assigning low scores to balls that are placed in dense regions, thereby delaying their evaluation until more information is known about the region.

Using this method, over 80% of the hexagons were found. The remaining hexagons were missed largely due to "cleanup" step of the algorithm. In images with densely packed hexagons, small errors in the first few rounds would propagate forward, such that there were few legal positions to place a new hexagon in the later rounds. The hexagon would nevertheless be placed too close to one found in a previous round, and both would be deleted from the list. Because of this, hexagons located in the very center of large clusters would often be missed. While several fixes were attempted, it soon became obvious that any solution using heuristics would be complicated.

As a result, neural networks were implemented in unison with the scoring algorithm. Neural networks are the computational analogues of biological brains. Networks of simulated neurons are trained with a set of inputs and expected outputs. The synaptic weights connecting each node in the network are then adjusted so as to minimize error. As a result, the neural network is able to "recognize" trends in the data that may not be obvious to a human analyst. The end goal is a network that can generalize, such that it can produce the correct output given input it was not trained on.

Neural networks were especially suited for finding hexagons because they work well with fuzzy data. While the scoring algorithm used hard-coded thresholds, neural networks use a series of synaptic weights that allow for error due to noise and other real-world factors. Because of this, neural networks are very efficient at recognizing patterns in image data, just like their biological counterparts.

The training data for the neural networks used in this program consisted of several thousand, 13×13 pixel bitmaps of hexagons, as identified by a human operator. Also included were samples of non-hexagons. Several networks were trained using the Python FFNet module, each analyzing a different channel of the image. The FORTRAN code generated by the Python module was then translated to IDL and included in the original program.

Several of these networks formed a committee. Each ball in the image would be passed through the committee, and, if enough of the networks agreed, the ball was added to the list of hexagon centers. While this method found virtually all of the hexagon centers, it also found many erroneous centers, often positioned in clumps. To correct for this, the same cleaning procedure as used in the scoring algorithm is implemented. The remaining hexagons are kept in a separate list, and those that were deleted are evaluated a second time by the scoring algorithm. Those with the highest scores are once again added to the list of centers.

In all, the program goes through three distinct segments while finding hexagons. First it uses only neural networks. Next, it uses neural networks aided by a scoring algorithm to reduce error. Finally, only the scoring algorithm is used. The results from all three steps are then combined in a manner that gives preference to the neural networks, as they were found to be the more accurate of the two methods.

III. RESULTS AND ANALYSIS

Combining these two methods yielded excellent results. On average, 98.5% of hexagons were found across usable data sets. Out these, the exact center was located 95% of the time. There were two data sets that did not perform well, as the program struggled to find 90% of the hexagons. This is due in part to the fact that the camera was placed considerably farther back in these two sets, and the lighting is significantly different. Given that there are similar data sets that performed

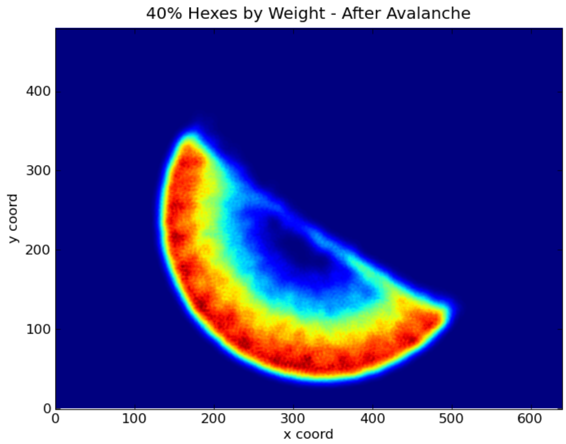

FIG. 3: Smooth binning, showing the segregation of the mixture. Taken from data spanning 70 avalanchess.

very well under the same algorithm, these under-performing images may be considered outliers. Furthermore, the sets in question are of limited use in analysis, due to the fact that they each consist of mostly all hexagons or all doubles.

The combination of the neural networks with the scoring algorithm represented the most efficient method of finding hexagons. Separately, each approach was mediocre. The neural networks were good at working with less than ideal data, whereas the scoring algorithm was not, due to its hard-coded parameters. On the other hand, the neural networks lacked global perspective, since there was no way to account for the number of expected hexagons per image or other constraints during training. This is where the scoring method was advantageous. By using both methods in parallel, the final result was optimized.

In order to quickly visualize trends in the resulting data, heat maps were generated. "Smooth" binning assigned a range of of scores to hexes, depending on their distance from the binning node. "Spot" binning only assigned scores to hexagons within a very limited distance of the node. Smooth binning thus emphasizes the density of a region. If many hexagons are packed into an area, the center of the area will accumulate a great amount of "heat." Spot binning, on the other hand, promotes frequency. If a hexagon was near a specific node in a large number of images, that node will have a higher score.

Various general trends were reinforced by analyzing the heat maps, and possible new correlations were identified. One behavior previously recognized was the segregation of grain sizes. After only a few avalanches, hexagons would gather along the bottom of the drum, while doubles would rise towards the middle-center. This is especially evident in the heat maps (see Figure 3). Warmer colors, representing a high frequency of hexagons in the region, cluster along the outer rim of the drum. Moving toward the center, a sharp boundary occurs at a constant radius. Looking at a single avalanche, this boundary is not as apparent.

Striation is a common property of granular materials, but one not immediately obvious in this experiment. Nevertheless, spires can be seen growing from the outer edge of the drum up towards the center, especially in the spot heat maps (see Figure 4, Appendix). This is caused by hexagons stacking on top one another in image after image.

A possible new trend shown by the heat map relates hexagon positions to the angle at which the mixture avalanches. Two heat maps were made; one using data from the five avalanches with the highest angles, and one with data from the five lowest. When compared, it was seen that the piles that avalanched at lower angles had greater concentrations of hexagons in the bottom right of the heap (Figure 5, Appendix). The mixtures with high angles of avalanche tended to have clusters of hexagons in the upper left of the drum (Figure 6, Appendix). Upon inspection, it appears that the avalanche often begins with the grains located in the upper left corner of the heap. Results from previous experiments show that the stability of hexagon and double mixtures is inversely proportional to the concentration of doubles. Therefore, it makes sense that a greater concentration of the more stable hexagons in the region responsible for starting avalanches might lead to a higher angle of collapse. Unfortunately, though this trend is visible in several concentrations, it is not strongly represented. Nevertheless, this trend is worthy of further investigation.

IV. FURTHER RESEARCH AND CONCLUSION

The updates to the program allow it to find individual grains, enabling in-depth analysis of the mixture and its grain configuration. In order to gain more useful results, several additional steps could be taken.

In order to generate more conclusive heat maps, more image data could be recorded. The current heat maps showing a possible correlation between hexagon positioning and angle of avalanche suffer from the small number of samples available to them. On average, there are 60 sample images per concentration. Taking 10% of the avalanches from the extremes of the angle distribution yields only six images worth of data per heat map. If several hundred additional images of one concentration could be recorded, any correlations evident in the resulting heat maps could be accepted with much more confidence.

An interesting extension of the project would involve using a high speed camera to capture many frames of a single avalanche. The program could then be amended to trace the paths of individual avalanches as they move through the mixture. This would shed light on where avalanches begin, and which grains are most affected. Because it is not feasible to record for any sizable amount of time using a high speed camera, and avalanches cannot be anticipated, this method would require some method of triggering the recording once an avalanche is detected. The current implementation of the hexagon identification program is not fast enough to be able to optically identify an avalanche in real time, therefore another method, such as a microphone trigger, would need to be used.

Furthermore, the current program is restricted to identifying hexagons only. In future experiments, it might be desirable to look at mixtures of different shapes, such as diamonds and doubles. Doing so would require more than a simple rewrite of the code, since it makes crucial assumptions based on the fact that it is identifying only hexagons. The combined neural network and heuristic approach, however, has been validated by the search for hexagons, therefore it could be applied to other shapes as well.

It has thus been shown that a reasonable method exists for identifying specific grains in images of granular mixtures. This allows for analysis of individual configurations, rather than the traditional study of the generic behaviors of a mixture as a whole. Possible trends were identified that deserve further investigation, and the use of heat maps was validated as a tool for visualization.

[1] For more information, see *Segregation and Stability of Granular Mixtures*, Swartz et al

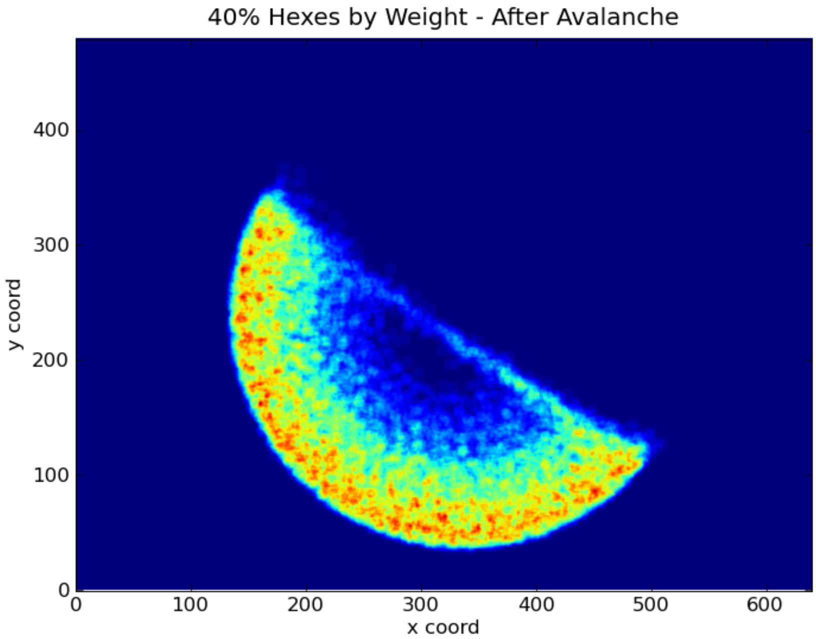

FIG. 4: Spot heat map showing striations. Note the bands of orange rising into the green area.

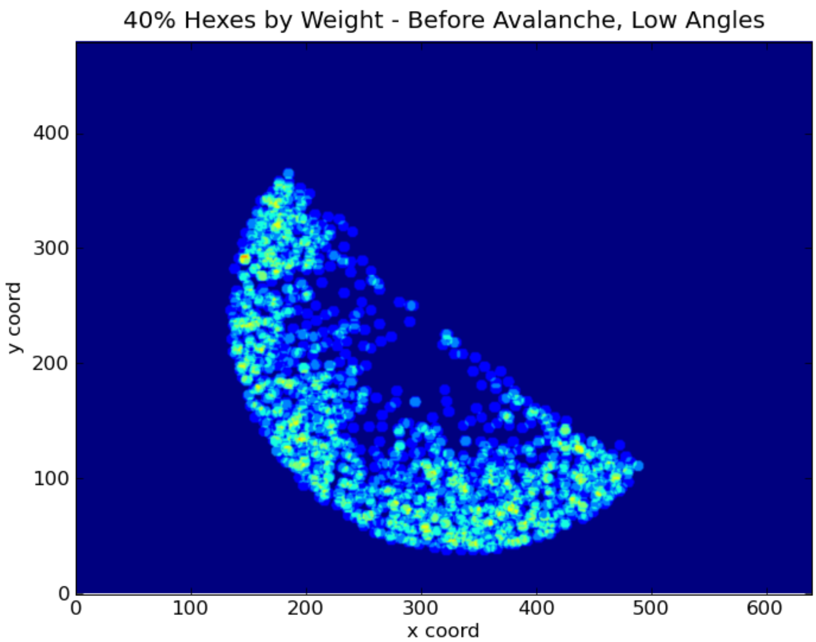

FIG. 5: A spot heat map using data from the five lowest angle avalanches in the 40% concentration. Note the cluster of hexagons in the bottom right.

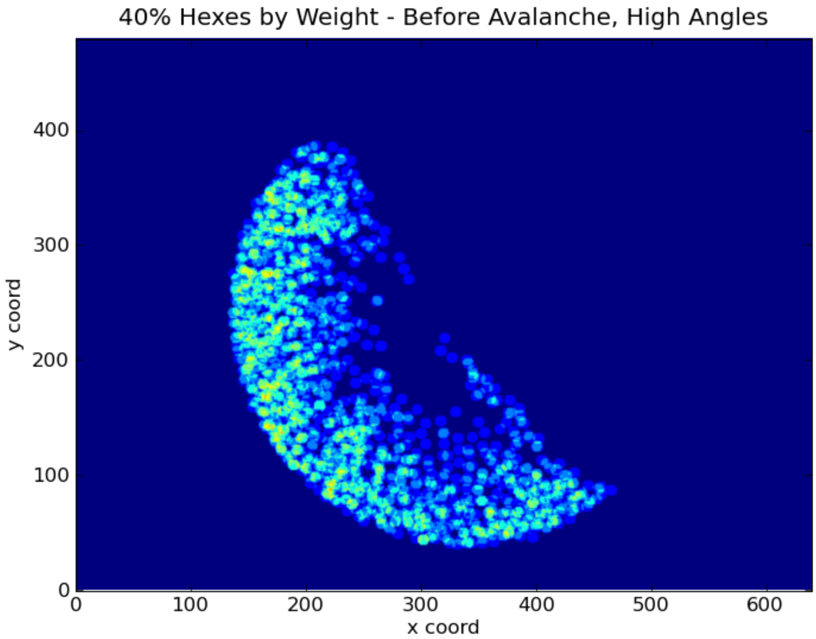

FIG. 6: A spot heat map using data from the five highest angle avalanches in the 40% concentration. In this case, the hexagons appear to cluster in the upper left region.