What is shape?

Characterizing particle morphology using genetic algorithms and deep generative models

A review by
Simon Guldager Andersen

UNIVERSITY OF COPENHAGEN

Granular Matter (2023) 25:2
https://doi.org/10.1007/s10035-022-01282-y

ORIGINAL PAPER

What is shape? Characterizing particle morphology with genetic algorithms and deep generative models

R. Buarque de Macedo[1] · S. Monfared[1] · K. Karapiperis[2] · J. E. Andrade[1]

Received: 8 December 2021 / Accepted: 5 September 2022

Abstract
Engineered granular materials have gained considerable interest in recent years. For this substance, the primary design variable is grain shape. Optimizing grain form to achieve a macroscopic property is difficult due to the infinite-dimensional function space particle shape inhabits. Nonetheless, by parameterizing morphology the dimension of the problem can be reduced. In this work, we study the effects of both intuitive and machine-picked shape descriptors on granular material properties. First, we investigate the effect of classical shape descriptors (roundness, convexity, and aspect ratio) on packing fraction ϕ and coordination number Z. We use a genetic algorithm to generate a uniform sampling of shapes across these three shape parameters. The shapes are then simulated in the level set discrete element method. We discover that both ϕ and Z decrease with decreasing convexity, and Z increases with decreasing aspect ratio across the large sampling of morphologies—including among highly non-convex grains not commonly found in nature. Further, we find that subtle changes in mesoscopic properties can be attributed to a continuum of geometric phenomena, including tessellation, hexagonal packing, nematic order and arching. Nonetheless, such descriptors alone can not entirely describe a shape. Thus, we find a set of 20 descriptors which uniquely define a morphology via deep generative models. We show how two of these machine-derived parameters affect ϕ and Z. This methodology can be leveraged for topology optimization of granular materials, with applications ranging from robotic grippers to materials with tunable mechanical properties.

Keywords Granular materials · Non-convex · Topology optimization · Deep generative models · Discrete element method · LS-DEM

Structure of the presentation

I. Setting the stage
II. Genetic algorithms – generally and specifically
III. Variational autoencoders – a very brief introduction
IV. Results and discussion
V. Questions?

Genetic algorithms are the main focus of this presentation

I. ~ Setting the stage

What we want: Predict the grain shape needed to give a granular material certain physical properties specified by us

How we'll do it:

→ Create many particle shapes with a genetic algorithm

→ Use these to train a VAE to construct a basis for grain shape

→ For each grain shape of interest, simulate the interaction of many identical grains to extract bulk properties

→ Understand the mapping between the basis and bulk material properties

II. ~ Genetic algorithms - Overview

- Stochastic global optimization scheme
- Biased towards better solutions, but allows for worse to avoid getting stuck in local minima
- Many hyperparameters – careful and patient tuning necessary

II. ~ Genetic algorithms - Stages

Initialize a population of N solutions (randomly or using prior knowledge)

At each iteration (not necessarily in this order):

1) **Fitness evaluation & Selection:** Evaluate the quality of each solution and select k solutions through a selection scheme. Discard the rest
2) **Crossover:** Produce N-k new solutions by combining the current k solutions through a crossover scheme
3) **Mutation:** Mutate each solution with probability p through a mutation scheme

Terminate when convergence criterion or max iterations are met

II. ~ Genetic algorithms – Selection stage

Fitness evaluation – *evaluate the quality of all solutions:*

- ‘Quality’ is quantified by the cost function
- Incorporate prior knowledge into the cost function, e.g. by adding penalty terms for non-physical solutions

Selection – *select k individuals through a selection scheme like e.g.:*

- Roulette wheel selection
- Tournament selection

II. ~ Genetic algorithms – Crossover stage

Crossover stage – *Produce N-k new solutions by combining the current k solutions through a crossover scheme:*

- *Pick two solutions at random and crossover with probability* p_c

Crossover scheme examples:

- Linear combinations of two or more solutions
- Swapping one or more components of two solutions

Swapping Example; crossover at the second index:

$$Parent\ 1 = (x_1, x_2, x_3, x_4, x_5), \quad Parent\ 2 = (y_1, y_2, y_3, y_4, y_5)$$

yielding

$$Child\ 1 = (y_1, y_2, x_3, x_4, x_5), \quad Child\ 2 = (x_1, x_2, y_3, y_4, y_5)$$

II. ~ Genetic algorithms – Mutation stage

Mutation stage – *Mutate each solution with probability* p_m *through a mutation scheme:*

- Perturb one or more components by a random value(s) drawn by some distribution
  - → Gaussian results in mostly small steps
  - → Lévy results in small steps combined with occasional big jumps

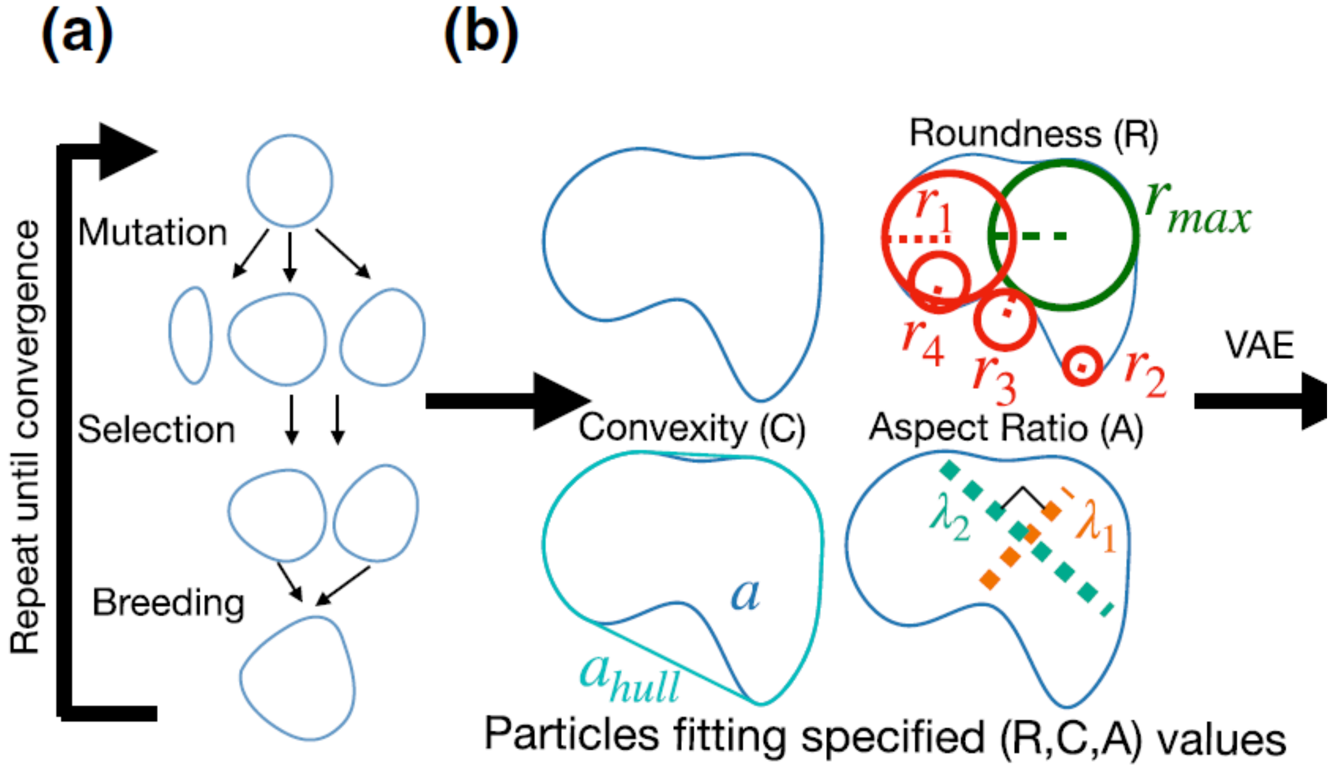
II. ~ Using GAs for particle generation

Which parameters are relevant to the shape of a 2D object?

$$R = \frac{\sum_{i=1}^{N} r_i}{r_{max}}$$

$$C = \frac{a}{a_{hull}}$$

$$A = \frac{\lambda_1}{\lambda_2}$$



II. ~ What does the cost function look like?

$$cost = (C - C_{target})^2 + (R - R_{target})^2 + (A - A_{target})^2 + 100 \cdot SC + 100\, SI + 100\, BN$$

$$\mathrm{SC} = \begin{cases} 1, & \text{if any corner is too sharp} \\ 0, & \text{otherwise} \end{cases}$$

$$\mathrm{SI} = \begin{cases} 1, & \text{if particle self} - \text{intersects} \\ 0, & \text{otherwise} \end{cases}$$

$$\mathrm{BN} = \begin{cases} 1, & \text{if points of opposite side are too close} \\ 0, & \text{otherwise} \end{cases}$$

II. ~ Generate particle with a given (R, C, A)

Initialize $N = 50$ particles, each one as 8 points uniformly distributed on an ellipse with aspect ratio A

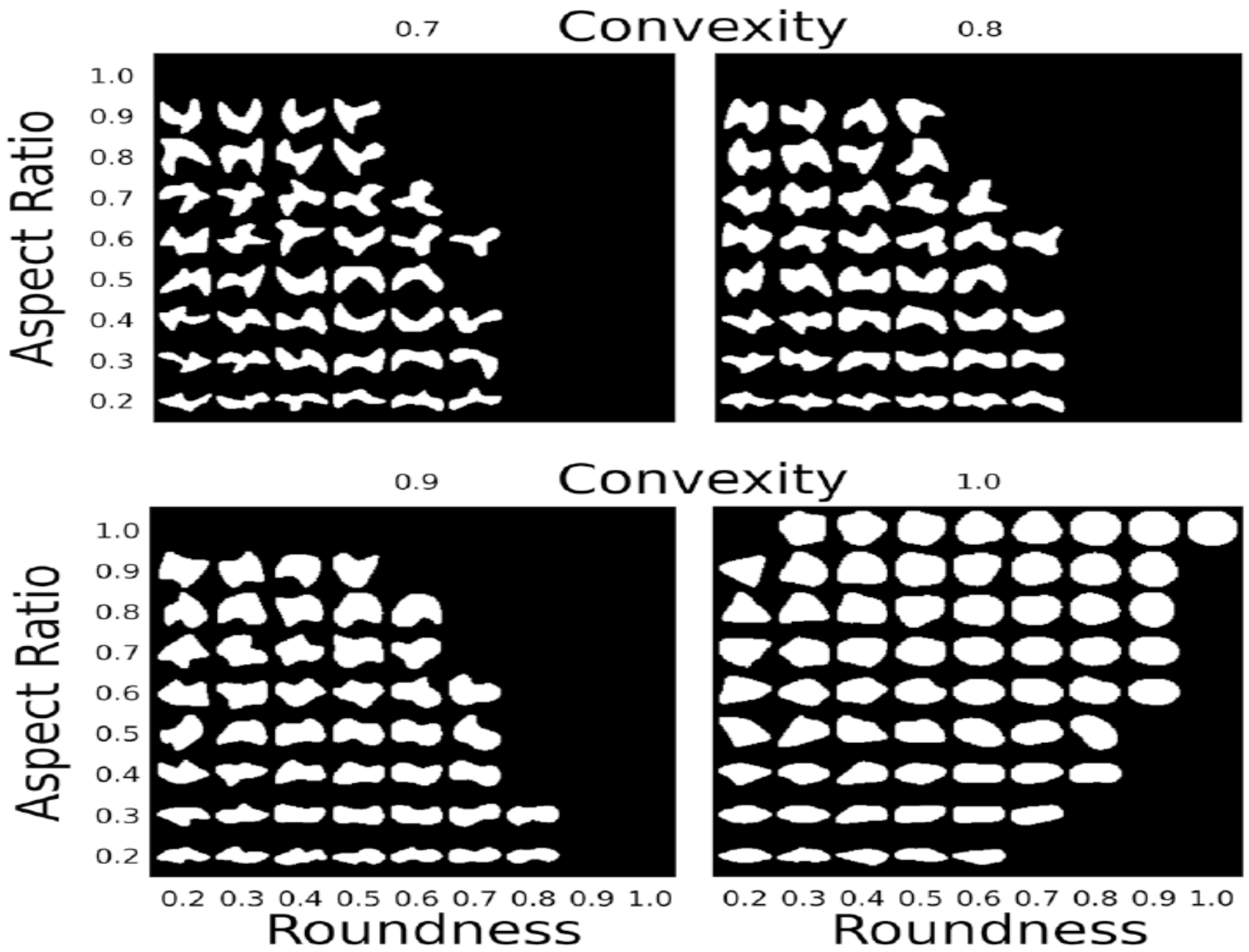
At each iteration:

- **Breed** k new solutions by choosing solution pairs at random with $p_c = 0.2$ and randomly swapping points.
- **Mutate** each particle point (r, θ) with $p_m = 0.5$ by drawing values from Gaussians specified by $(\mu_r = 0, \sigma_r = 1)$ and $(\mu_\theta = 0, \sigma_{theta} = 0.05)$
- **Select** the best $\sim \frac{N+k}{2}$ solutions by tournament selection. Duplicate the winners until the population size is 50

Terminate when the minimum cost of a solution is less than 0.0005

II. ~ Examples of generated particles

The point of everything so far has been to create 10,000 unique images as training data for the VAE



III. ~ Variational autoencoders – *very* briefly

- Deep generative model
- Dimensionality reduction
- Latent space distribution

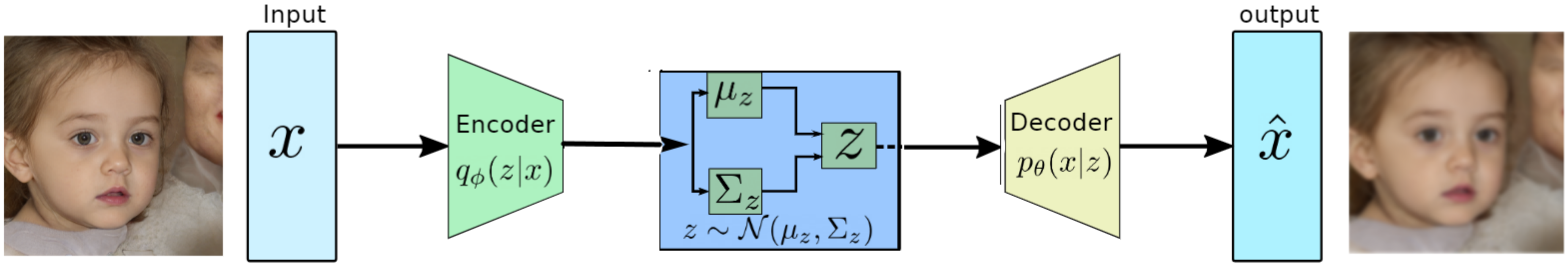
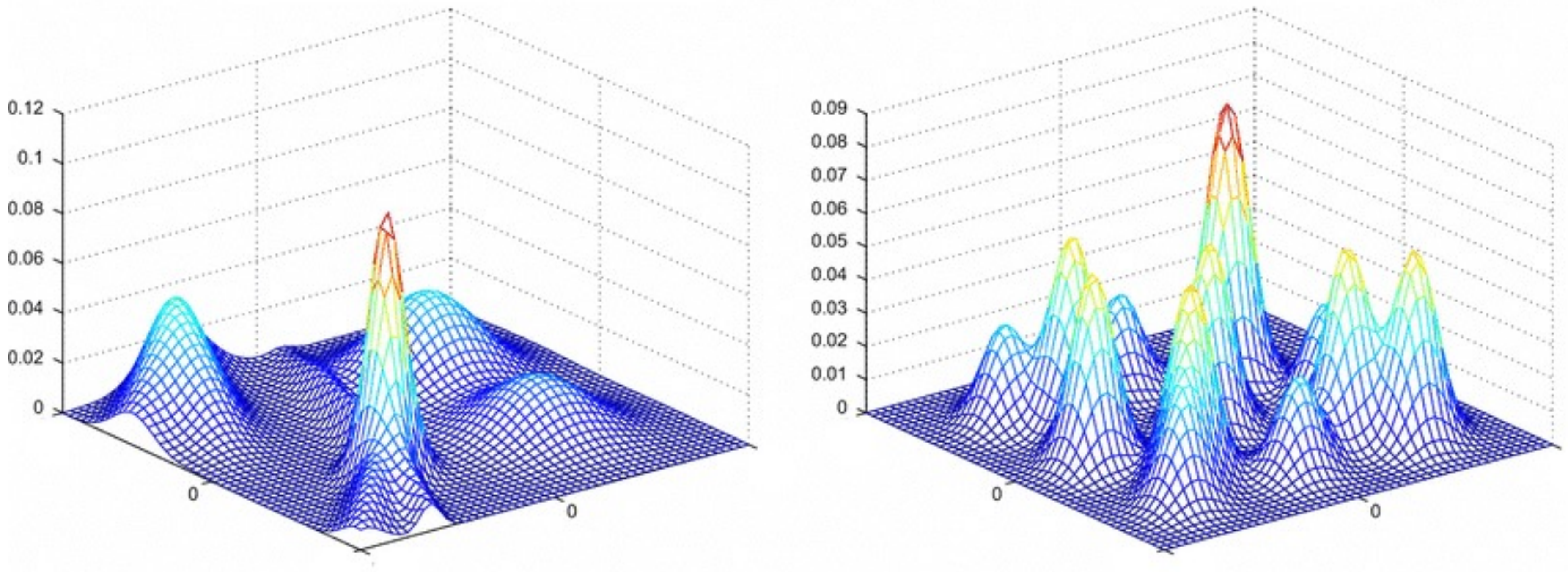


Image by Sunil Yadav

III. ~ Variational autoencoders – *very* briefly

Possible latent distributions of a 2 dimensional latent space



(a) (b)

III. ~ Constructing a space of possible shapes

- Generate 10,000 unique images of particle shape as training data
- Assume that training data roughly 'maps out' true shape space
  → i.e. no important regions of shape space without images
- During traning, the VAE learns to interpolate between shapes
- After training, the VAE maps any latent vector to a unique shape
  → In this sense, we have built a space of possible grain shapes

III. ~ Training a VAE with the GA-shapes

- Trained on the 10,000 images of particle shapes
- A 20-dimensional latent space enough for accurate reconstruction
- Unphysical shapes when $z_i \notin [-4,4],\ i = 1, \ldots, 20$
- Using just 10 subdivisions per dimension → 10^{20} unique shapes

IV. ~ Results and discussion

- Having constructed space for possible grain shapes, mapping to bulk properties can in principle be obtained
  - → Many simulations needed to establish relationships
  - → Not given that each dimension has a simple physical interpretation

Questions & comments?