Vol. 13, No. 4: 53–60

Calculating Center of Mass in an Unbounded 2D Environment

Linge Bai and David Breen
Drexel University

Abstract. We study the behavior of simple, 2D, self-organizing primitives that interact and move in an unbounded environment to create aggregated shapes. Each primitive is represented by a disk and a unit point mass. In order to compare the aggregated shape produced by the primitives to other shapes, the centers of mass of the two shapes must be aligned. We present an algorithm for calculating the center of mass (COM) for a set of point masses that are distributed in an unbounded 2D environment. The algorithm calculates the centroid for each coordinate component separately by forming two "orthogonal" tubes, calculating a center of mass in 3D for each tube and then projecting the 3D COM back onto the tubes, in order to produce the 2D COM of the points.

1. Introduction

We are studying automated shape composition based on self-organizing primitives. We have developed a method for discovering local interactions that direct the primitives to aggregate into a user-defined shape [Bai et al. 08a]. The primitives are randomly placed in an unbounded 2D environment and allowed to assemble by following local field gradients. The primitives' environment has toroidal topology, since the top edge of the computational arena is connected to the bottom edge, and the left edge is connected to the right edge. This allows the primitives to move and interact with other primitives across arena boundaries. This type of computational configuration is also described as having periodic boundary conditions. A central component of our self-organization method is a genetic programming process [Koza 1992]

1086-7651/08 $0.50 per page

that requires that we quantify the similarity of the aggregated shape with the user-specified shape [Bai et al. 08b]. In order to compute the similarity metric, the shapes must first be placed in overlapping areas of the computational environment. Therefore, we make the center of mass (COM) of the aggregate coincident with the COM of the user-defined shape in the center of the shape's 2D Cartesian (image) space. In this paper, we present an algorithm for calculating the COM for a set of point masses in an unbounded 2D (toroidal) environment.

In general, the center of mass C of a set of point masses is calculated by the weighted average of all the points as in the equation

$$C = \frac{\sum m_i X_i}{\sum m_i}, \tag{1}$$

where m_i is the mass of point i and X_i is its location. Since our point masses exist in an unbounded environment, the problem is complicated by the lack of a proper origin for the environment's coordinate system. The specific problem we are solving involves finding the center of mass for n unit point masses in a rectangle with toroidal topology, where the origin of the finite rectangle is in the lower-left corner. The COM is needed in the coordinate system of the rectangle. Equation (1) cannot be used to provide a solution, since the collection of points may be aggregated across the environment's boundaries. For example in Figure 1 (left), the points on the top-left edge are close to points near the top-right corner and bottom-right edge, because of the toroidal connectivity. Similarly, in Figure 2 (left) the points along the top edge of the image are close to the points along the bottom edge of the image. Simply applying Equation (1), using the 2D Cartesian coordinates of the points in

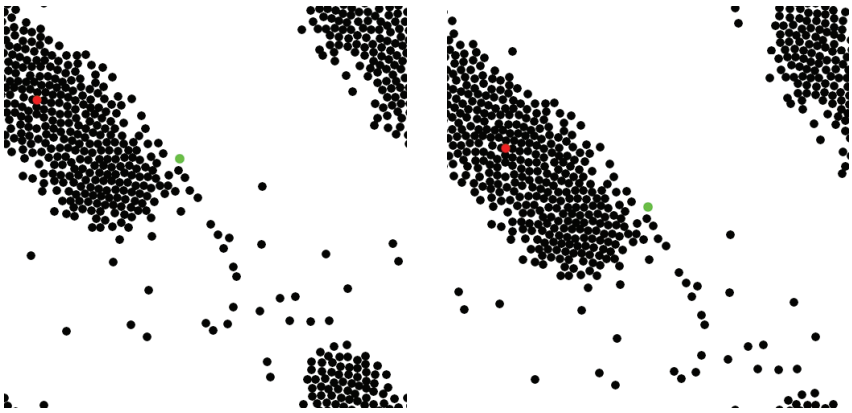

Figure 1. Example 1. Left: original image. Right: incorrect centered result.

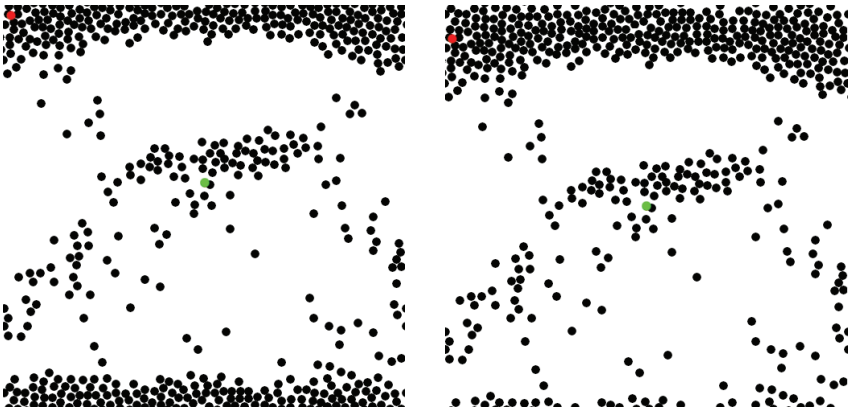

Figure 2. Example 2. Left: original image. Right: incorrect centered result.

the images, produces incorrect solutions, as seen in Figures 1 (right) and 2 (right). In these images, a green dot is placed at the center of mass produced by the above method. The COM calculated by our new method is represented by a red dot, which can be found in the upper-left portion of the images. When we compare aggregation results, the computed COM (green dot) is translated to the center of the image, with the pixels being rolled across edge boundaries. The desired result should place the aggregate in the center of the image. Using the 2D Cartesian coordinates of the points along with Equation (1) does not produce the desired outcome.

One existing method, based on SLERP, computes a weighted spherical average [Buss et al. 01]. The weighted average is expressed similar to Equation (1) but on the surface of a torus. The COM of the point masses is the point that minimizes the sum of the weighted distance between the COM and each point mass. However, the calculation of distance on a toroidal surface is complex, and the outcome of the minimization may have multiple local solutions. Therefore, we propose a new, simpler algorithm to solve the COM problem in an unbounded 2D environment.

2. Algorithm Description

In order to properly calculate the COM of 2D point masses distributed in an environment with periodic boundary conditions, the masses are first mapped onto 3D tubes. A COM calculation is performed in three dimensions and then projected back into two dimensions. The algorithm calculates the centroid for each coordinate component separately by forming two tubes and calculating a center of mass in 3D for the points on each tube. The 3D COM is then

projected back onto the tubes. The location of the projected COM in the coordinate system of the tube surface is the 2D COM of the points.

The location of a point mass in the 2D rectangle is represented by the 2D Cartesian coordinate (i, j), with values ranging from $(0, 0)$ to $(i_{\max}, j_{\max})$. The algorithm constructs two "orthogonal" tubes from the 2D rectangle. We use the term *orthogonal* because one tube is formed by connecting the left edge of the rectangle with the right edge, and the other tube is formed by connecting the top edge with the bottom, conceptually creating two tubes rotated 90 ° from each other. The first tube ($\mathcal{T}_i$) is created by connecting the $i = 0$ edge of the rectangle with the $i = i_{\max}$ edge. The second tube ($\mathcal{T}_j$) is created by connecting the $j = 0$ edge of the rectangle with the $j = j_{\max}$ edge. The process of forming the tubes transforms the primitives' locations from 2D to 3D. The 3D Cartesian coordinate $\mathbf{X}_k \equiv (x, y, z)$ denotes the location of the kth point mass in three dimensions during these interim calculations. The 2D-to-3D transformation for points on tube $\mathcal{T}_i$ is defined by

$$\begin{aligned} x = r_i \cos(\theta_i), \qquad y = j, \qquad z = r_i \sin(\theta_i), \\ r_i = \frac{i_{\max}}{2\pi}, \qquad \theta_i = \frac{i}{i_{\max}} 2\pi. \end{aligned} \tag{2}$$

The 2D-to-3D transformation for points on tube $\mathcal{T}_j$ is defined by

$$\begin{aligned} x = i, \qquad y = r_j \cos(\theta_j), \qquad z = r_j \sin(\theta_j), \\ r_j = \frac{j_{\max}}{2\pi}, \qquad \theta_j = \frac{j}{j_{\max}} 2\pi. \end{aligned} \tag{3}$$

Two different tubes are needed in order to calculate each coordinate of the COM. Tube $\mathcal{T}_i$ is used to calculate the i-coordinate, because once transformed to cylindrical coordinates, i becomes the boundless θ_i-coordinate. Tube $\mathcal{T}_j$ is used to calculate the j-coordinate, because j is then mapped to θ_j.

Once the unit point masses have been mapped onto one of the tubes, the 3D center of mass of these transformed points is calculated:

$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{X}_k.$$

The 3D COM $\bar{\mathbf{X}}$ is then projected back onto each tube. We calculate just one of the COM coordinates from each tube. The coordinate $\bar{i}$ is calculated from $\bar{\mathbf{X}}$ for tube $\mathcal{T}_i$:

$$\theta_i = \operatorname{atan2}(-\bar{z}, -\bar{x}) + \pi, \qquad \bar{i} = \frac{i_{\max}}{2\pi} \theta_i. \tag{4}$$

The coordinate $\bar{j}$ is calculated from $\bar{\mathbf{X}}$ for tube $\mathcal{T}_j$:

$$\theta_j = \operatorname{atan2}(-\bar{z}, -\bar{y}) + \pi, \qquad \bar{j} = \frac{j_{\max}}{2\pi} \theta_j. \tag{5}$$

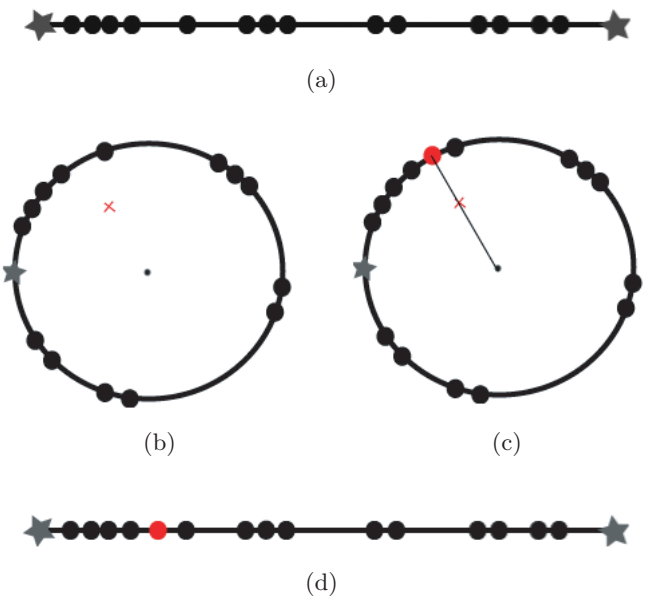

Figure 3. Applying the COM algorithm in 1D.

After this calculation, point $(\bar{i}, \bar{j})$ is the center of mass of the original 2D point masses in the 2D Cartesian coordinate system of the rectangle. The function atan2(y, x) is a Unix math function that computes the principal value of the arc tangent of y/x, using the signs of both arguments to determine the quadrant of the return value. It returns a value between $-\pi$ and π. The inputs to atan2 are negated, and the output is incremented by π in order to map the output of atan2 to the ranges of θ ($[0, 2\pi]$), defined in Equations (2) and (3).

A degenerate case occurs when the 3D center of mass lies on the central axis of a tube. In this case, the projection of the 3D point onto the surface of the tube is not well-defined. In Equations (4) and (5), the degenerate case occurs for $atan2(0, 0)$. This (unlikely) situation can be detected, and the output θ value can simply be defined as 0. The 3D point is then projected to one of the edges of the 2D rectangle. However, after rolling the 2D rectangular image, the projected 2D point will always be at the center of the rolled image.

A 1D example of the algorithm is presented in Figure 3. Point masses are given along a line representing a connected circular environment (Figure 3(a)). The ends of the line are conceptually connected and marked by a star. In order to calculate the COM of the masses, the line is formed into a circle.

The COM is calculated using the new 2D positions and marked by a small red "x" (Figure 3(b)). The intermediate 2D COM is projected back onto the circle and marked by a red disk (Figure 3(c)), providing the 1D COM in the coordinates of the line (Figure 3(d)). The degenerate case of 1D COM calculation occurs when the 2D COM (the red "x" in Figure 3(b)) overlaps with the center of the circle. The 2D COM can then be projected to either end of the 1D line.

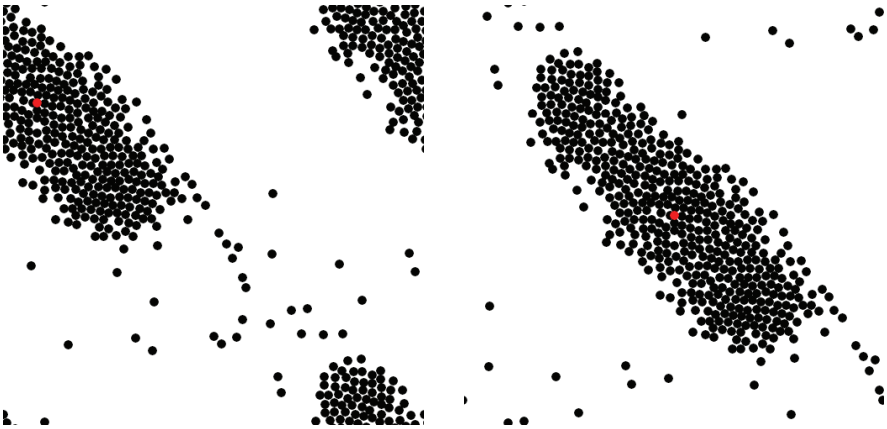

Figure 4. Example 1. Left: original image. Right: desired centered result.

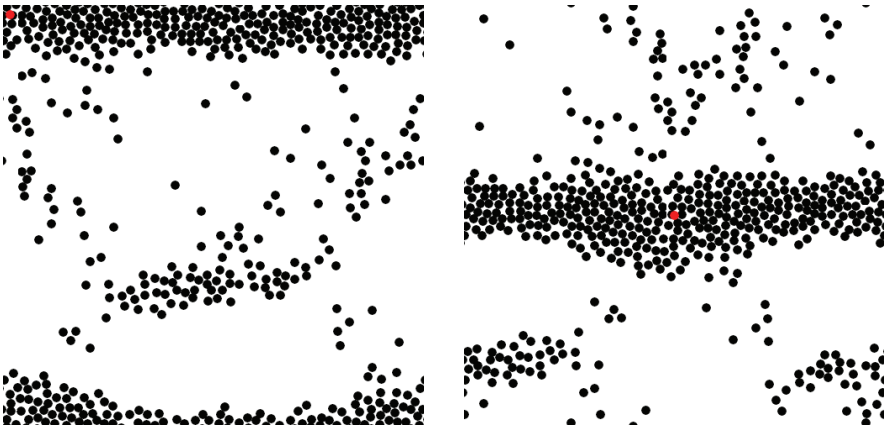

Figure 5. Example 2. Left: original image. Right: desired centered result.

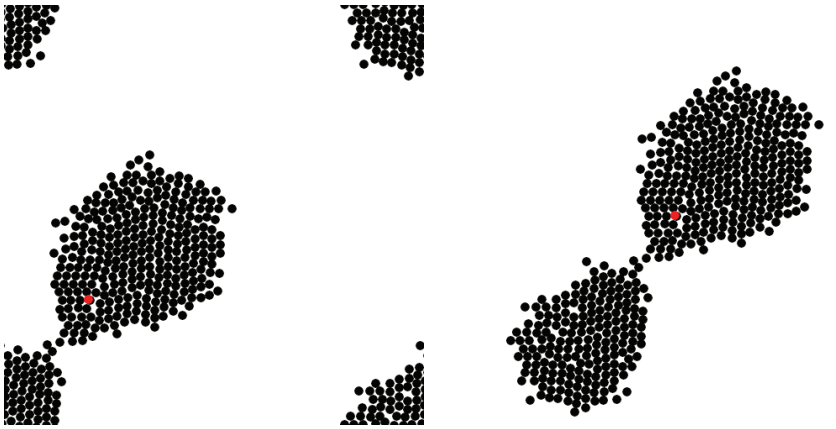

Figure 6. Example 3. Left: original image. Right: desired centered result.

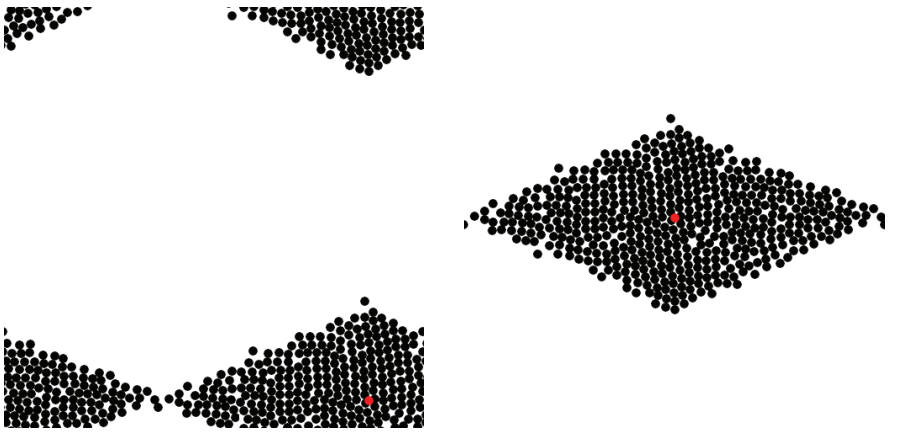

Figure 7. Example 4. Left: original image. Right: desired centered result.

3. Examples

We present four examples from our self-organization simulations that use the COM algorithm to shift aggregates, formed in an unbounded 2D environment, to the center of an image. We roll the COM $(\bar{i}, \bar{j})$ to the center of the image $(i_{\max}/2, j_{\max}/2)$ as seen in Figure 4 through Figure 7. In all of these examples, the main aggregated object crosses several boundaries of the image rectangle. In the left images, the COM of the point set is displayed with a red dot. In the right images, the COM (red dot) is shifted to the center of the image

rectangle, with the same shift applied to all of the points. If the shift makes an individual point cross an image edge, the point is placed on the opposite side of the image, i.e., the image is rolled.

We have found the COM algorithm to be effective and robust. It has been employed to calculate the center of mass for approximately 100,000 diverse test cases and has always produced satisfactory results in practice.

Acknowledgments. This research has been funded by National Science Foundation grant CCF-0636323.

References

[Bai et al. 08a] L. Bai, M. Eyiyurekli, and D. E. Breen. "Self-Organizing Primitives for Automated Shape Composition." In *Proc. IEEE International Conference on Shape Modeling and Applications*, pp. 147–154. Los Alamitos, CA: IEEE Press, 2008.

[Bai et al. 08b] L. Bai, M. Eyiyurekli, and D. E. Breen. "Automated Shape Composition Based on Cell Biology and Distributed Genetic Programming." In *Proc. Genetic and Evolutionary Computation Conference*, pp. 1179–1186. New York: ACM Press, 2008.

[Buss et al. 01] S. R. Buss and J. P. Fillmore. "Spherical Averages and Applications to Spherical Splines and Interpolation." In *ACM Transactions on Graphics (TOG)* 20:2 (2001), 95–126.

[Koza 1992] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA: The MIT Press, 1992.

Web Information:

http://jgt.akpeters.com/papers/BaiBreen08/

Linge Bai, Dept. of Computer Science, Drexel University, 3141 Chestnut St., Philadelphia, PA 19104 (lb353@cs.drexel.edu)

David Breen, Dept. of Computer Science, Drexel University, 3141 Chestnut St., Philadelphia, PA 19104 (david@cs.drexel.edu)

Received May 9, 2008; accepted January 29, 2009.