

Boids Algorithms

Author: Erlanda Miko Prasetya
Teacher: Muhammad Qomaruz Zaman, S.T., M.T., Ph.D.
Class name: Algoritma dan Komputasi
Github and Code can Access in,
[Boids Algorithms \(GitHub\)](#)
[Matlab Code \(GitHub\)](#)

1. Pendahuluan

Algoritma *Boids* merupakan model perilaku kolektif yang dikembangkan oleh Craig Reynolds pada tahun 1986 untuk meniru gerakan kawanan burung, ikan, atau kelompok agen lainnya [1]. Prinsip dasar algoritma ini adalah bahwa setiap partikel (*bird*) bergerak berdasarkan aturan lokal tanpa adanya koordinasi pusat. Dari interaksi sederhana antar agen, dapat muncul perilaku kolektif yang kompleks seperti pembentukan formasi atau pengelompokan alami.

Dalam simulasi ini, digunakan sejumlah partikel N yang bergerak pada bidang dua dimensi. Setiap partikel memiliki posisi $\mathbf{p}_i = [x_i, y_i]$ dan kecepatan $\mathbf{v}_i = [v_{x,i}, v_{y,i}]$. Posisi dan kecepatan partikel diperbarui setiap interval waktu Δt berdasarkan tiga komponen utama:

1. **Separation**: menghindari tabrakan dengan tetangga terdekat.
2. **Cohesion**: bergerak menuju pusat massa tetangga.
3. **Alignment**: menyesuaikan arah kecepatan dengan tetangga.

2. Parameter Simulasi

Parameter utama yang digunakan dalam simulasi adalah sebagai berikut:

Parameter	Simbol	Keterangan
Jumlah partikel	$N = 50$	Jumlah agen atau partikel
Ukuran area	$L = 100$	Rentang koordinat posisi
Langkah waktu	$\Delta t = 0.1$	Interval waktu simulasi
Kecepatan maksimum	$v_{\max} = 2$	Batas kecepatan partikel
Radius pengaruh	$R_b = 10$	Jarak maksimum interaksi
Koefisien kohesi	$k_c = 0.2$	Bobot gaya <i>cohesion</i>
Koefisien separasi	$k_s = 0.1$	Bobot gaya <i>separation</i>

Table 1: Parameter yang digunakan dalam simulasi Boids

3. Aturan Perilaku Boids

3.1 Separation

Aturan *separation* bertujuan untuk menghindari tabrakan antar partikel. Setiap partikel i menghasilkan gaya tolak terhadap tetangga j yang berada dalam radius pengaruh $r < R_b$. Vektor gaya *separation* dapat dituliskan sebagai:

$$\mathbf{V}_{\text{sep},i} = k_s \sum_{j \in \mathcal{N}_i} \frac{-(\mathbf{p}_j - \mathbf{p}_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|^2}$$

dengan:

- \mathbf{p}_i : posisi partikel ke- i ,
- \mathcal{N}_i : himpunan tetangga partikel ke- i dengan jarak $< R_b$,
- k_s : konstanta *separation*.

Gaya ini mengarahkan partikel untuk menjauh dari tetangga terdekat secara proporsional terhadap jarak kuadratnya.

3.2 Cohesion

Aturan *cohesion* membuat partikel bergerak menuju pusat massa dari tetangganya. Tujuannya adalah menjaga agar kelompok partikel tetap menyatu. Persamaan matematisnya sebagai berikut:

$$\mathbf{V}_{\text{coh},i} = k_c \left(\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{p}_j - \mathbf{p}_i \right)$$

dengan k_c adalah koefisien kohesi.

3.3 Alignment

Komponen *alignment* berfungsi agar arah kecepatan partikel i menyesuaikan dengan rata-rata arah kecepatan tetangganya. Dirumuskan sebagai:

$$\mathbf{V}_{\text{ali},i} = k_a \left(\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{v}_j - \mathbf{v}_i \right)$$

dengan k_a merupakan konstanta pengaruh *alignment*.

4. Implementasi dalam MATLAB

Potongan kode berikut menunjukkan implementasi sederhana dari algoritma Boids untuk $N = 50$ partikel menggunakan gaya *separation* dan *cohesion*.

```
1 clear; clc; close all;
2
3 % ---- Parameters ----
4 N = 50;
5 L = 100;
6 DT = 0.1;
7 MAX_SPEED = 2;
8 BINDING_RADIUS = 10;
9 COHESION = 0.2;
10 SEPARATION = 0.1;
11
12 % ---- Initialization ----
13 P = (2 * L) * rand(N, 2) - L;
14 V = 2 * rand(N, 2) - 1;
15
16 % Normalize initial velocities
17 speed_mag = sqrt(sum(V.^2, 2));
18 V_unit = V ./ speed_mag;
19 V = V_unit .* min(speed_mag, MAX_SPEED);
20
21 % ---- Graphics ----
22 h = plot(P(:, 1), P(:, 2), 'o', ...
23         'MarkerSize', 5, ...
24         'MarkerFaceColor', [0.2 0.8 1], ...
25         'MarkerEdgeColor', [0 0.5 1]);
26 set(gca, 'Color', [0, 0, 0]);
27 axis equal;
28
29 % ---- Main loop ----
30 while true
31     V_new = zeros(N, 2);
32     for i = 1:N
33         R = P - P(i,:);
34         dist = sqrt(sum(R.^2, 2));
35         idx = find(dist > 0 & dist < BINDING_RADIUS);
36
37         if isempty(idx)
38             continue;
39         end
40
41         % Separation
42         V_separation = sum(-R(idx,:) ./ dist(idx).^2, 1) * SEPARATION;
43
44         % Cohesion
45         CoM = mean(P(idx,:), 1);
46         V_cohesion = (CoM - P(i,:)) * COHESION;
47
48         % Combine
49         V_temp = V(i,:) + V_cohesion + V_separation;
50         sp = norm(V_temp);
51         if sp > MAX_SPEED
52             V_temp = V_temp * (MAX_SPEED / sp);
53         end
54         V_new(i,:) = V_temp;
55     end
56
57     V = V_new;
58     P = P + V * DT;
59
60     % Update graphics
61     set(h, 'XData', P(:, 1), 'YData', P(:, 2));
62     xlim([-L, L]); ylim([-L, L]);
63     drawnow limitrate;
64 end
```

Listing 1: Kode Simulasi Algoritma Boids

5. Analisis Hasil

Pada awal simulasi ($t = 12.3$ s) sebagaimana ditunjukkan pada Gambar 1, partikel-partikel masih tersebar secara acak di seluruh area simulasi. Hal ini menunjukkan bahwa gaya-gaya sosial (*separation* dan *cohesion*) belum sepenuhnya menyeimbangkan pengaruh awal dari kecepatan acak masing-masing partikel. Interaksi antar partikel masih bersifat lokal dan belum membentuk pola kolektif yang jelas.

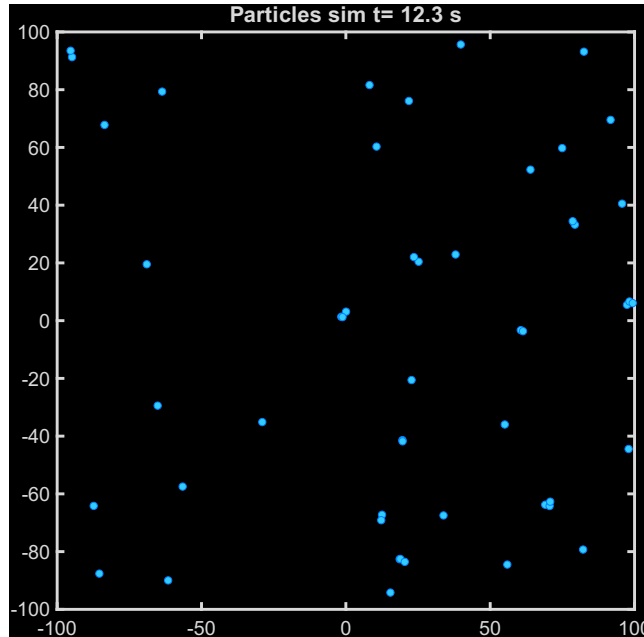


Figure 1: Distribusi posisi partikel pada waktu $t = 12.3$ s.

Seiring berjalannya waktu, pada $t = 306.2$ s (Gambar 2), terlihat bahwa partikel-partikel mulai membentuk kelompok-kelompok yang relatif stabil. Pola ini merupakan hasil keseimbangan dinamis antara gaya *separation* yang mencegah tabrakan dan gaya *cohesion* yang menjaga agar partikel tetap berdekatan. Pada tahap ini, sistem telah mencapai keadaan *self-organization* di mana setiap partikel mempertahankan jarak optimal terhadap tetangganya, menunjukkan perilaku kolektif menyerupai gerombolan (*flocking*).

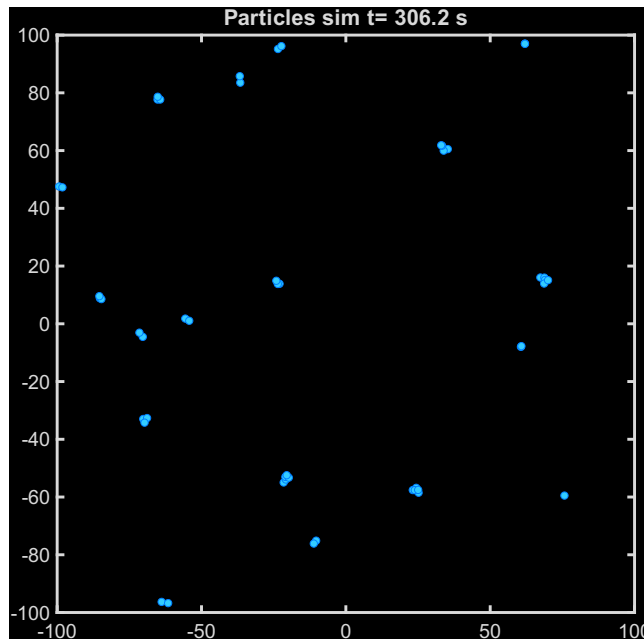


Figure 2: Formasi kelompok partikel pada waktu $t = 306.2$ s.

Secara umum, dapat diamati bahwa proses transisi dari distribusi acak menuju formasi terorganisir terjadi secara bertahap. Peran parameter *SEPARATION* dan *COHESION* sangat penting dalam menentukan jarak antar partikel dan kestabilan pola gerakan.

References

- [1] C. W. Reynolds, *Flocks, Herds, and Schools: A Distributed Behavioral Model*, Computer Graphics, vol. 21, no. 4, pp. 25–34, 1987.
- [2] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, *Collective memory and spatial sorting in animal groups*, Journal of Theoretical Biology, vol. 218, no. 1, pp. 1–11, 2002.
- [3] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, *Novel type of phase transition in a system of self-driven particles*, Physical Review Letters, vol. 75, no. 6, pp. 1226–1229, 1995.
- [4] R. Eberhart and J. Kennedy, *A new optimizer using particle swarm theory*, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, 1995.