

# Perimeter Compression in self-healing swarms

Brockway M, Eliot N, Kendal D,  
Hexham University  
Cronkley Campus  
Department of Computer Science

January 26, 2021

## Abstract

Perimeter Compression is a technique where by a void reducing effect can be added to a basic swarming algorithm. The affect is dependant upon perimeter identification and is controlled by applying two factors to the existing swarming formula. One to the cohesion calculation and the other to the repulsion calculation.

## 1 Introduction

Perimeter compression is a technique that creates a “pull” effect between perimeter agents. It is dependant upon perimeter agent identification as discussed by Eliot et. al. in the Alife Paper [3].

The aim of the algorithm is to reduce the spacing between perimeter-based agents by reducing the repulsion field (Figure. 1) and increasing the cohesion affect on perimeter agents.  $S_b$  is the sensor field.  $O_b$  is the obstacle field.  $C_b$  is the cohesion field.  $R_b$  is the repulsion field. The implementation involves introducing two controlling factors;  $k_{cpc}$  (Cohesion Perimeter Compression) which increases the cohesion vector ( $C_b \rightarrow k_{cpc}C_b$ ) and  $k_{rpc}$  (Repulsion Perimeter Compression) which reduces the size of the repulsion field ( $R_b \rightarrow k_{rpc}R_b$ ) on the inter-perimeter agents.

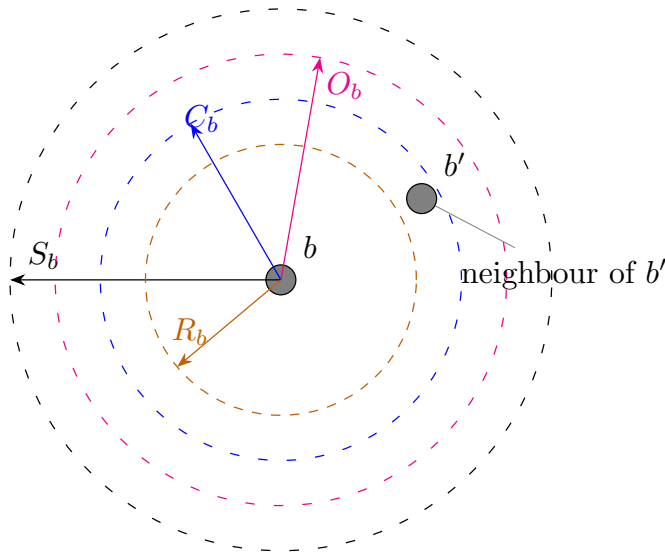


Figure 1: Agent Fields

## 2 Resultant Vector Calculation

In the Original work by Eliot et. al. the resultant vector of an agent was calculated using Equation 1. Where  $k_c, k_r, k_d, k_o$  are weighting factors for the summed vectors associated with each interaction. The new algorithm requires each individual agent to have a variation to the vector generated inside each calculation based on the perimeter status of the agent and each neighbour. The equation therefore needs to have an additional weighting factor applied to specific agents within the cohesion and repulsion vector calculations ( $k_{cpc}, k_{rpc}$ ) as shown is Equation 2).

$$v(b) = k_c v_c(b) + k_r v_r(b) + k_d v_d(b) + k_o v_o(b) \quad (1)$$

$$v(b) = k_c v_c(b, k_{cpc}) + k_r v_r(b, k_{rpc}) + k_d v_d(b) + k_o v_o(b) \quad (2)$$

the effects of the addition of these additional weighting factors can be seen in Figure 2. The metric used in producing the graph is based upon the inter-agent magnitudes [2].

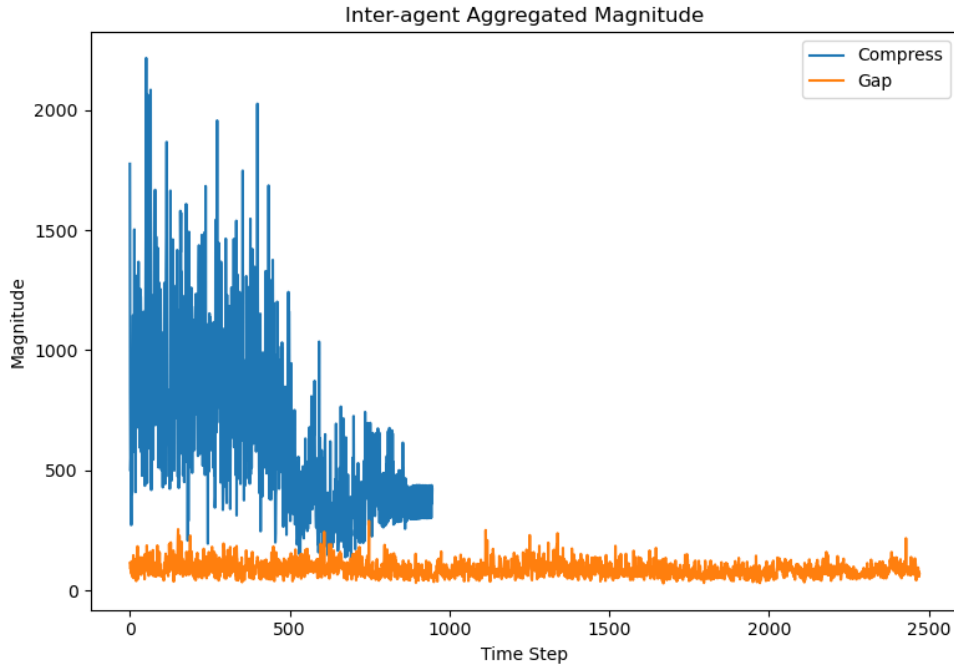


Figure 2: Compression Effect based on Magnitude change

## 3 Repulsion

The repulsion component of an agent's movement is calculated from its interaction with its neighbours  $nbr(b)$  that are within the agent's ( $b$ ) repulsion field ( $R_b$ ) (Eq. 3) or from any agent in the swarm ( $S$ ) that are within the agent's ( $b$ ) repulsion field ( $R_b$ ) (Eq. 4). The resultant set will be the same.

$$R(b) = \{b' \in S : \|bb'\| \leq R_b\} \quad (3)$$

Given that the repulsion field is within the cohesion field (neighbours) then the repulsion set can also be identified as a subset of those neighbours where  $nbr(b)$  (Eq. 8) is the set of all of the neighbours of  $b$  [1]:

$$R(b) = \{b' \in nbr(b) : \|bb'\| \leq R_b\} \quad (4)$$

### 3.1 Compression repulsion

**Assumption 1**  $k_{rpc} \leq 1$

To reduce the repulsion effect, the control factor ( $k_{rpc}$ ) is applied to an agent's repulsion field if both itself and its neighbour are perimeter agents. Where  $per()$  returns an agent's perimeter status. An agent is identified as a perimeter agent using the technique shown by Eliot et.al. in [3]

Therefore if the condition of both agents being a perimeter is met ( $per(b) \wedge per(b')$ ), where  $per()$  returns *true* if an agent is on the perimeter and or *false* if it is not, and  $b'$  is within repulsion field distance multiplied by the compression factor, or both agents are not both perimeter agents ( $\neg(per(b) \wedge per(b'))$ ) and  $b'$  is within the 'normal' repulsion field ( $R_b$ ) then an agent is part of the repulsion set (Eq. 5 or 6). Equation 6 being the least expensive.

$$\begin{aligned}
 R(b) = \{b' \in S : \\
 & (per(b) \wedge per(b') \wedge \|bb'\| \leq k_{rpc}R_b) \\
 & \vee \\
 & (\neg(per(b) \wedge per(b')) \wedge \|bb'\| \leq R_b) \\
 & \}
 \end{aligned} \tag{5}$$

$$\begin{aligned}
 R(b) = \{b' \in nbr(b) : \\
 & (per(b) \wedge per(b') \wedge \|bb'\| \leq k_{rpc}R_b) \\
 & \vee \\
 & (\neg(per(b) \wedge per(b')) \wedge \|bb'\| \leq R_b) \\
 & \}
 \end{aligned} \tag{6}$$

The effect of Equations 5 and 6 is that the perimeter-based agents will be closer together before a repulsion vector is generated.

**Important:** The repulsion vector that is generated is based upon the  $R_b$ , the full repulsion field, and not the reduced field. This is to reduce potential agent collisions.

The final calculation for the repulsion having generated the set  $R(b)$  is shown in Equation 7 .

$$v_r(b) = \frac{1}{|R_b|} \left( \sum_{b' \in R_b} \left( 1 - \frac{\|b'\|}{R_b} \right) b' \right) \tag{7}$$

## 4 Cohesion

**Assumption 2**  $k_{cpc} \geq 1$

The cohesion component of an agent is calculated in a similar way to the repulsion in that it is dependent upon the proximity of neighbours  $nbr(b)$  (Eq. 8). The determining factor for an agent being affected is determined using the cohesion field ( $C_b$ ). The cohesion set  $nbr(b)$  is equivalent to  $C(b)$  (Eq. 9).

$$nbr(b) = \{b' \in S : \|bb'\| \leq C_b\} \tag{8}$$

$$C(b) \equiv nbr(b) \tag{9}$$

## 4.1 Compression cohesion

The compression effect on the cohesion of the perimeter agents is that any perimeter-perimeter relationships should be re-enforced using the cohesion compression quotient ( $k_{cpc}$ ). This is achieved by identifying the relationship within the cohesion calculation.

$$v_c(b) = \frac{-1}{|\mathcal{C}_b|} \left( \sum_{b' \in \mathcal{C}_b} b' \right) \quad (10)$$

=====

PROBLEM

=====

I'm having a problem with my notation (more than usual here) I cannot work out how to specify that if the relationship between  $b$  and  $b'$  is perimeter to perimeter ( $per(b) \wedge per(b')$ ) I need to multiply the resultant vector by  $k_{cpc}$  to increase the cohesion between those two agents?

=====

## 5 Conclusions

From the initial simulations it is possible to show that the technique is able to successfully remove voids and surround an obstacle as shown in the video <https://youtu.be/3eY1vvq0JWo>.

## References

- [1] N. Eliot. *Methods for the Efficient Deployment and Coordination of Swarm Robotic Systems*. University of Northumbria at Newcastle (United Kingdom), 2017.
- [2] N. Eliot, D. Kendall, and M. Brockway. A new metric for the analysis of swarms using potential fields. *IEEE Access*, 6:63258–63267, 2018.
- [3] N. Eliot, D. Kendall, A. Moon, M. Brockway, and M. Amos. Void reduction in self-healing swarms. In *Artificial Life Conference Proceedings*, pages 87–94. MIT Press, 2019.