# 鳥群演算法 Boids 與其應用

## 一、 動機

生物模擬一直都是我很感興趣的題目，包括課堂中所提到的蟻群演算法。最早碰到鳥群演算法(Bird-oid objects algorithm，簡稱 Boids，亦稱作 flocking algorithm，以下簡稱 Boids)是我在做一個水下探險遊戲的時候。那時我想要有一大群的魚類在海中游動，然而我的電腦並不允許這麼大量的移動軌跡同時計算，一般的粒子模擬則沒有預期的真實，於是我開始摸索各種方法，這才發現原來早在將近 40 年前，就有人遇到了一樣的問題。

電腦科學家克雷格·雷諾茲(Craig Reynolds)發明的 Boids 演算長久以來被廣泛用於電腦特效上；而隨著近年無人載具興起，這項發明也逐漸被運用在此領域。

這讓我好奇這項仿生技術的起源，以及它是如何被應用在遊戲與無人機科技上，還有它未來的發展會是如何。

## 二、 內容介紹

### (一) 起源

1986 年，電腦科學家克雷格·雷諾茲(Craig Reynolds)為了模擬鳥群飛行而編寫了第一代的 Boids 演算法。1987 年，Boids 演算法被用於動畫片《Stanley and Stella in: Breaking the Ice》中，為 Boids 首次被使用在動畫片中；同年，雷諾茲也發布了他的論文《Flocks, Herds, and Schools: A Distributed Behavioral Model》，其中即包含對此演算法的探討。Boids 演算法隨後也被用於 Batman Returns(1992)、The Lion King(1994)、StarWars 系列、Lord of the Ring 三部曲以及鐵達尼號(1997)等片中，用於動物群體、軍隊等的群集模擬。克雷格·雷諾茲在 1998 年因




圖 1：動畫《Stanley and Stella in: Breaking the Ice》（1987）宣傳海報。圖片來源：維基百科，https://en.wikipedia.org/wiki/Stanley_and_Stella_in:_Breaking_the_Ice

「對電影製作中 3D 電腦動畫的發展作出的開創性貢獻」獲得奧斯卡科技成果獎。

## (二) 生物原理

Boids 演算法的目的在模擬鳥群飛行的方式。雷諾茲在論文中提及了鳥群的湧現行為，也就是說鳥群所展現的型態並非由一個中心的思考控制，而是由許多微小的、「與鳥群本身性質不同」的個體(也就是鳥)所構成，鳥群只是這個集合之中的成員相互作用而展現的型態。

而 2008 年的論文 Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study，則進一步的說明群體生物間的「取樣」標準是使用拓樸距離(最近的幾隻)而非絕對距離。



圖 2：群飛的椋鳥。圖片來源：John Holmes，〈The flock of starlings acting as a swarm.〉，Wikimedia Commons（CC BY-SA 2.0），https://en.wikipedia.org/wiki/File:The_flock_of_starlings_acting_as_a_swarm._-_geograph.org.uk_-_124593.jpg

## (三) 技術轉化過程

在模擬上，與其消耗大量資源計算每隻鳥的軌跡，電腦可以透過在鳥與鳥之間施加簡單的規則，再限制總體的方向。Boids 會將個體的行為分為三個部分：分離、對齊、聚集。

1. 分離（Separation）：每個個體會盡量避免與其他個體發生碰撞。
2. 對齊（Alignment）：每個個體的速度會受附近個體的平均速度方向影響。
3. 聚集（Cohesion）：每個個體會朝附近個體的平均位置移動。

每個個體會在固定範圍內取樣附近個體(也可能取樣定量個體或「視野內個體」，因算法而異)，再依據既有的權重衡量實際的移動方向。

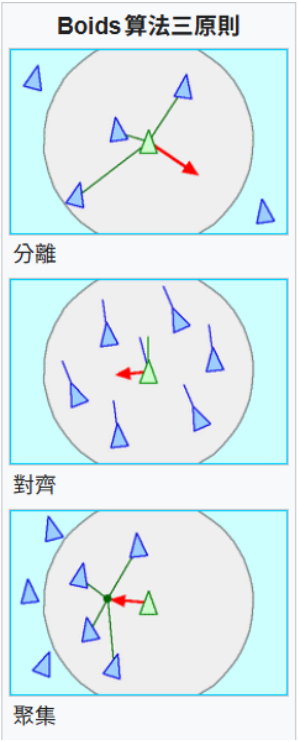



圖 3：Boids 算法三原則。圖片來源：維基百科，https://zh.wikipedia.org/zh-tw/Boids

## (四) 應用：階層化 RTS 尋路

在千禧年前至 2000 年代初期，Boid 的主要運用仍維持在影視特效與電腦模擬上，而其中一個他所解決的問題就是即時戰略(Real-Time Strategy, RTS)遊戲中的尋路。




圖 4：即時戰略遊戲中用 Boids 控制單位移動的除錯畫面展示了算法三原則。圖片來源：jdxdev，〈Boids for RTS〉，https://www.jdxdev.com/blog/2021/03/19/boids-for-rts/

在非玩家角色數量較少的遊戲中，每個實體可以各自利用 A-star 等尋路算法找尋完整的路徑。然而，戰略遊戲往往需要玩家統馭千軍萬馬(畢竟那是他們最大的賣點)，若是對每個單位都分別進行尋路，這即便是現在的 CPU 也吃不消。

因此，對於由大量、相同目標的單位所構成的軍團，遊戲會在上層使用尋路演算法規劃大致路徑，再使用 Boids 來控制每個單位的行為，這讓對於近代電腦來說也十分吃力的工作能流暢地在 20 年前的電腦上執行。

## (五) 應用：無人機群動

同時，他高效率的運算很快就吸引了機器人學領域的注意。在 Flocking for multi-agent dynamic systems: algorithms and theory 這篇 2006 年的論文中，研究人員利用 Boids 建立了更嚴謹的數學模型，來解決機器人群動常見的問題，包括三維避障、網絡穩定性以及可擴展性。

而在 2010 年代，許多包含實機測試的論文也逐漸證明了 Boids 類演算法在無人機科技的實用性。2018 年的論文 Optimized flocking of autonomous drones in confined environments 中，研究人員延伸了 Boids 的分離、對齊、聚集方法，並實機測試了 30 台無人機，機群能夠避障，並在短暫分離後會合。

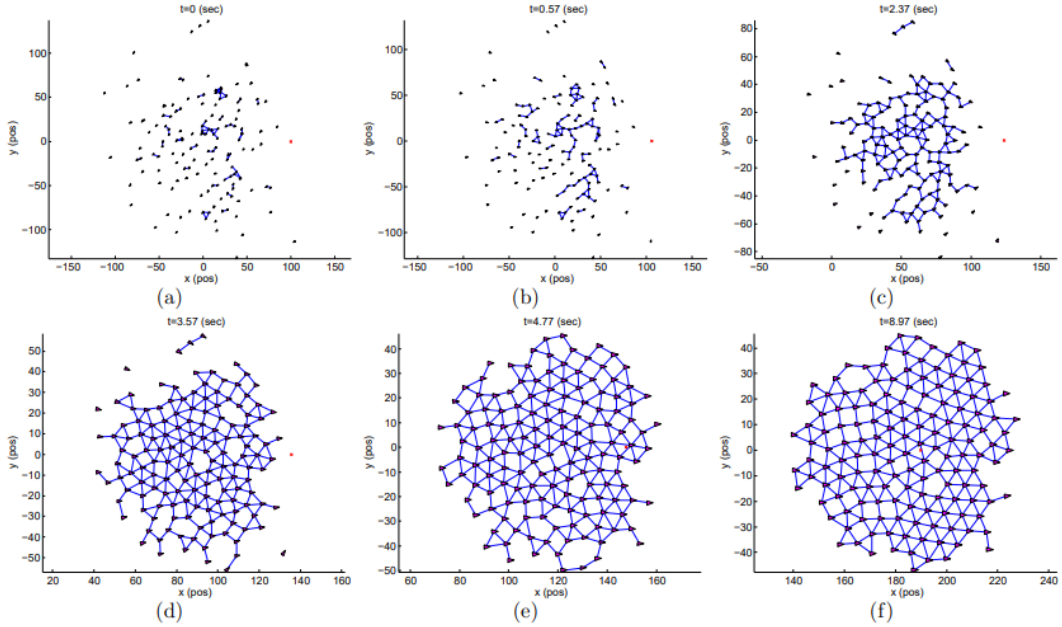



Figure 11: 2-D flocking for $n = 100$ with a dynamic topology.

圖 5：100 個 Agent 的 2D Flocking 模擬，從 (a) 到 (f) 隨時間收斂成網格隊形。圖片來源：Olfati-Saber (2006), Figure 11

隨著群集無人機技術逐漸被廣泛應用於娛樂、工業、救災及軍事等領域，Boid 的應用也不再限縮於視覺效果。

2025 年的論文 ARBoids: Adaptive Residual Reinforcement Learning With Boids Model for Cooperative Multi-USV Target Defense 使用了效率較高的 Boids 做為底層控制，並加上強化學習訓練的 Agent 做複雜化控制，來控制防禦用的無人船艇，可見 Boid 在軍事上的應用還在逐漸擴展。

# 三、 心得反思

生物的群體行為是令人稱奇的。他們沒有如我們一般的智慧(先假設他們沒有對我們有所隱瞞)，卻能夠在本能與學習建立的規則下，構成這樣完整且美麗的律動。而人類自詡為星球上最聰明的生物，卻也因此從來無法得到這樣的和諧。「湧現」一詞從來都不局限於鳥群，這樣的現象存在於一切之中，如原子與分子、分子與細胞、細胞與人之間，沒有兩者是相同的，而人與人群亦然。人群從不存在方向，但人人都有自己的。也因此，我們試著用社會規範對齊彼此、用情緒渲染聚集彼此，然後依著我們的本性分離。

但我要說的並不是人類又哪裡比不上動物了。即便人與人之間的關係注定如此，我們在想方設法交織出一個秩序的過程中，每個個體都在不斷的走自己的

道路。在獨立與個體之間拉扯，這就是人類的宿命，也是人生之所以有趣的地方。

做為一個群體生物，我們固然還有很多要向動物學習。他們利用極簡的原則，就能達到極高效的組織。然而人類的做法會有些許不同，個體的獨特是我們不會犧牲的變數。

# 四、 參考資料


1. Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques (pp. 25-34).
2. Carlson, W. E. (2017). Computer graphics and computer animation: A retrospective overview. The Ohio State University.
3. Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., Viale, M., & Zdravkovic, V. (2008). Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. Proceedings of the National Academy of Sciences, 105(4), 1232–1237. https://doi.org/10.1073/pnas.0711405105
4. Reynolds, C. W. (1999). Steering behaviors for autonomous characters. In Proceedings of the Game Developers Conference (pp. 763-782).
5. Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Transactions on Automatic Control, 51(3), 401-420. https://doi.org/10.1109/TAC.2005.864190
6. Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E., & Vicsek, T. (2018). Optimized flocking of autonomous drones in confined environments. Science Robotics, 3(20), eaat3536. https://doi.org/10.1126/scirobotics.aat3536
7. Tao, J., Shen, T., Zhao, D., & Zhang, F. (2025). ARBoids: Adaptive residual reinforcement learning with boids model for cooperative multi-USV target defense. arXiv. https://doi.org/10.48550/arXiv.2502.18549

# 五、 生成式 AI 使用聲明

工具名稱 1：Gemini (2026)

使用範圍：輔助進行報告大綱的邏輯結構優化、反思的脈絡順暢化、演算法專有名詞（如拓樸距離、湧現）的銜接微調、整理文獻 APA 引用格式、整理總體生成式模型使用、文法錯別字與句型流暢度修正。

工具名稱 2：Claude (2026)

使用範圍：圖標、圖片說明、圖片來源及圖片版權標示的整理及校閱

資訊驗證方法：報告內文提及之所有歷史案例（如《Stanley and Stella》動畫、奧斯卡科技成果獎獲獎年份）及學術論文，均由本人透過 Google Scholar 與學術資料庫核對原始文獻，確認脈絡無誤後，由本人親自撰寫完成全文。