CGIT Background Essay - Flocking

Although prevalent, traditional keyframe style animation is not always appropriate for capturing certain scenarios. This project will focus on one such situation: a 3d flocking simulation. Birds flock in an incredibly majestic and coordinated fashion, swooping, and turning in unison. Research from zoologist Wayne Potts showed that birds don't simply copy a leader or neighbours (Potts, 1984). Rather, they observe the approaching "mauver wave" as it moves through the flock and anticipate their own change in direction to coincide with the wave reaching their position.

While some earlier solutions to simulating this behaviour exist (Girard and Amkraut, 1990), by far the most influential is that of Craig Reynolds' "Boids" (a combination of bird and droid). This method was first described in his paper "Flocks, Herds, and Schools: A Distributed Behavioral Model" (Reynolds, 1987). Reynolds proposes an expansion upon a traditional particle system, wherein each particle represents a bird, with a distributed behavioural model allowing the aggregate group to display convincing flock-like behaviour. Reynolds' algorithm has stayed relevant to this day, despite its age. As such, it is pertinent to spend time delving into his solution, with some details glossed over for brevity.

Particle systems (Reeves, 1983) have traditionally been used to model elements like fire and smoke (although more involved methods now exist). Particles are stored as a collection, where each has individual behaviour which alters its state (colour, position, speed etc.). Reynolds' system builds on this, with each Boid particle holding not just a position but an orientation, as well as more complex behavioural logic that depends on the system, not just an individual particle.

The velocity of each Boid is stored as a vector, encapsulating both speed and orientation. This velocity is updated on each frame in accordance to three rules: Collision Avoidance, Velocity Matching and Flock Centring (Figure 1).

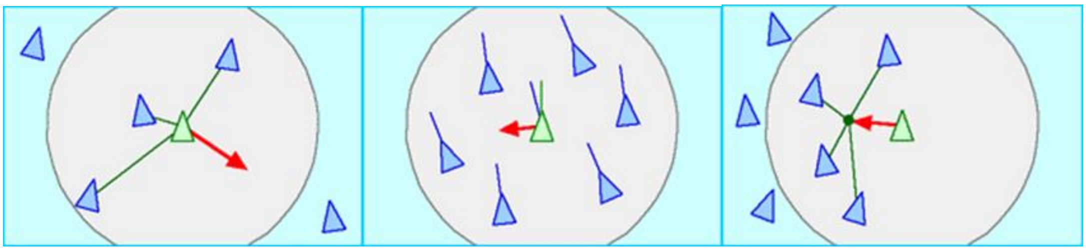

Figure 1: diagrams of Collision Avoidance, Velocity Matching and Flock Centring respectively (2022)

Collision avoidance checks the positions of neighbouring agents in the system and prompts the Boid to change its velocity to the opposite direction if neighbours stray too close. Velocity matching checks the velocity of nearby Boids and suggests an aggregate of these velocities so that the Boid may match speed and orientation with the surrounding flock. Finally, Flock centring prompts the Boid to change its velocity to steer towards the centre of neighbouring agents. This rule ensures that Boids on the edge of the flock do not fly away. For all 3 of these rules the agent is only aware of its neighbours within a defined region, to mimic the constrained awareness of a real bird. In addition, the inverse square of the distance is utilised to prioritise the nearest neighbours. Each of these rules returns a vector suggestion for the Boids motion. User defined parameters control the weight of each of these rules, which are used to scale the vectors. To avoid the rules cancelling each-other out they are added to the resultant vector in a user defined order of priority.

This resultant vector represents the acceleration of the agent for a time step of one frame. Using the equation $A = \Delta V / \Delta T$, we can rearrange to get the change in velocity as $\Delta V = A \Delta T$. Upon each frame update we set $t = 1$ (unless using an explicit time step), and so the resultant acceleration vector (A) is simply the change in velocity for that frame and is added to the current velocity vector. If the maximum acceleration (a specified parameter) is exceeded, the latest vector to be added to the result is trimmed to fit.

To ensure Boids steer away from other objects in the scene, Reynolds implements two methods of obstacle avoidance. The first, a "force field" model, constitutes of a series of repulsion force fields emanating from obstacles out into space, which are used to calculate avoidance acceleration for the Boids. Reynolds notes some drawbacks to this method, most notably the fact that head on collisions will merely be slowed rather than avoided. The second method, "steer-to-avoid", is a more natural method. Only obstacles directly in front of the agent are considered. From the local perspective of the agent, the edge of the object is found, and the Boid is steered at a point one body length away from this edge. This method is more complex to implement but negates the drawbacks of "force fields" by ensuring Boids are steered away from obstacles, as well as only considering objects in their line of sight (rather than in the periphery).

When all these components are combined, Boids correctly avoid obstacles, while flying around in an aggregate direction while still operating independently. Solitary Boids or smaller groups join to make larger flocks and can split apart to avoid obstacles (as in reality). Despite the relative algorithmic simplicity, the results are largely very convincing when compared to real birds. Much like Herbert Simons famous musings on an ant wandering a beach (Warner and Simon, 1969), here simple behavioural rules when applied to a complex environment (in this case a flock of neighbours and surrounding obstacles) give rise to complex behaviour. This behaviour (flocking movement) would be hard to capture using traditional animation methods and would be tedious to create individually for each bird. In providing a feasible alternative that is still used to this day, Reynolds' work is a resounding success. However, it would be remiss to imply there are no issues with the approach.

In the time since Reynolds' paper, many alternative methods to collision avoidance (specifically between agents rather than obstacles) have been proposed, notable examples are discussed in the paper "Algorithms for Microscopic Crowd Simulation: Advancements in the 2010s" (Toll and Pettré, 2021). The Collision Avoidance rule (also known as the separate rule) of Reynolds boids is a force-based model in which neighbouring agents cause a repulsive force to be applied. The most notable iteration on this is so called "collision prediction". First proposed in the paper "A predictive collision avoidance model for pedestrian simulation." (Karamouzas, heil, Van Beek and Overmars 2009), this model considers not just if a collision is imminent based on neighbour distance, but whether current velocities will cause agents to collide in the future. It introduces a "time to collision" calculated between pairs of neighbours (this will equal infinity if no collision is to occur in future with present velocities). The value of this time to collision then modifies the repulsive force received by an agent (with several different methods existing to do this modification (Zanlungo, Ikeda and Kanda, 2011). The popular 'Universal Power Law' method (Karamouzas, Skinner and Guy, 2014) improves collision avoidance, at the cost of additional computation to calculate the time to collision. This has the potential to become quite a hindrance as agent numbers increase and makes this method less suitable for real time flocking.

Velocity-based collision avoidance (Paris, Pettré and Donikian, 2007) is an extension of the previous predictive approach. Here each agent considers a set of potential velocities and evaluates them against criteria including future collisions before selecting the best option.

By pre examining various velocities, more complex and convincing behaviour can be exhibited than is possible using the force based model described previously. However quite clearly, this is at the cost of a far greater computational cost due to the increased number of predictions which must be made. This makes this method even less suitable for real time flocking with large numbers of agents.

A more recent innovation is vision-based avoidance, specifically retina based algorithms (Ondřej, Pettré, Olivier and Donikian, 2010). Here it is attempted to give agents a more realistic view of their environment. Instead of exact information about the position and velocity of neighbours, a virtual field of view for each agent is constructed and they make inference purely based on pixel information in this field. For the purposes of flocking simulation, it is possible that this method would more accurately simulate a birds view of its environment and hence result in more natural motion. However, this approach seems needlessly complex for this purpose, and a similar limitation of an agents vision could likely be achieved in a much simpler manner by limiting the radius and viewing angle of a neighbourhood vision sphere. In addition, this method once again requires much heavier computation than would be ideal for real time.

Unlike a real flock, the number of members acceptable to a Boids flock is constrained. This is indicative of the most troublesome issue: complexity. The big O complexity of the flocking algorithm is O(N^2), meaning the computational work done (and hence time to execute) grows in proportion to the square of the number of Boids in the flock. This results in a sharp cut-off in speed as more Boids are added. Reynolds suggests a few solutions. The first, to have a separate processor for each Boid assumes a number of processors that is unreasonable in most cases. A similar approach was later trailed by Lorek and White (1993), who gave a fixed number of Boids to each processor. They found that to simulate N Boids with P processors, each processor must communicate with all other (P-1) processors. This resulted in an efficiency of O(N2/P), rather than the O(N) Reynolds suggests.

Another solution proposed by Reynolds is to separate the flock into “bins” based on position, with Boids only checking neighbouring bins. This is still O(N2) but reduce the size of N, improving speeds. The paper "Parallel Simulation of Group Behaviours" (Zhou and Zhou 2004) takes this bin approach and combines it with the method used by Lorek and White. Here Zhou and Zhou partition the virtual space into bins and assign each one to a processor. This way each processor only needs to receive information from the processors managing neighbouring partitions. They show that this greatly improves speed (figure 2). Reynolds later described a similar method specifically for ps3 hardware (Reynolds, 2006). Even without parallel processing, the bin partitioning method appears highly effective and is the most feasible speed improvement to be implemented within this flocking project.

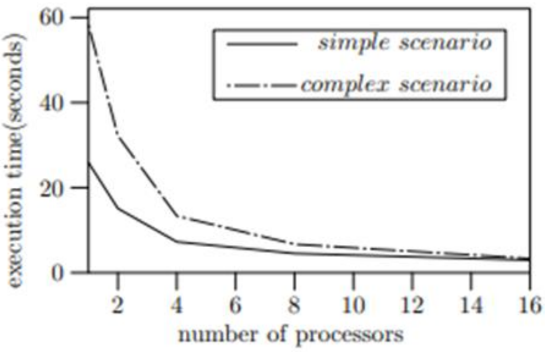

Figure 2: results from the Zhou and Zhou paper showing an exponential decrease in speed with the number of processors (Zhou and Zhou 2004).

Speed is not the only issue. An agent-based approach, while less work, leads to less explicit control over the output motion compared to traditional animation methods. This can result in a lot of fiddling with parameters to get the desired behaviour which can be especially frustrating for artists. Work has since been done to add additional control, some notable papers are discussed briefly here. The paper “Constrained 3D Flocking Behaviour” (Hernandez and Welborn 2011) proposes that artificial potential-fields (APF) (a method Reynolds used for collision avoidance) are used to direct agents, in addition to Reynolds’ three rules. A user can define 3d waypoints which map a path for agents to follow. Additionally, a boundary parameter constrains the outward spread of the flock along this path. Both are useful additions to Reynolds’ method, adding more control for artists.

Another paper, “Swarming Behaviour Using Probabilistic Roadmap Techniques” (Bayazıt, Lien and Amato, 2005), suggests that Reynolds’ method is sub optimal when applied to a more complex environment (examples given are a city scene or crowded room). To combat this, they borrow a route planning method often used in early robotics (Nilsson 1984). A network of prospective paths through the scene is computed during pre-processing, and then used as reference to steer the agents. Having global knowledge of the environment allows for more complex and natural motion than simply having agents assess their immediate environment. However, the requirement for pre-processing is an obvious hindrance.

An issue with the agent-based models presented by Reynolds is that it is sometimes hard to encapsulate edge case or subtle behaviours with the broad flocking rules. In recent years research has been done on data driven approaches which use real world data to either evaluate or produce agent behaviour more similar to that of real-world examples.

Early data driven methods (Lerner, Chrysanthou and Lischinski, 2007) and (Lee, Lee and Hong 2007) hold a database of crowd information based on real life examples. Each entry is a short segment of motion for a specific agent, with information about its surroundings and neighbours. At runtime, each agent finds an entry in the database most similar to its current situation and uses the velocity stored there. This is effective at reproducing real world behaviours but does not guarantee collision free paths for agents. In addition, input data quality and size have a drastic effect on motion quality and speed. Deep learning methods have since been used to try to alleviate some of these issues by creating a behavioural model using neural networks (Alah, Goel, Ramanathan, Robicquet, Fei-Fei and Savarese 2016) (Lee and Lee 2018). These can be used to create a more abstract behaviour model which can adapt to new situations more effectively and quickly than exactly replicating movement from a database.

Data driven methods have also been used to evaluate the accuracy of simulated crowd motion (Karamouzas, Sohre, Hu, and Guy 2018) (Guy, Chhugani, Curtis, Dubey, Lin and Mamocha 2010). Simulated motion is compared with real world data, this can score correctness more accurately than by eye. An issue with using real world data for evaluation rather than behaviour creation is that you assume a ground truth to your underlying data set. Outside of very simple scenarios there are likely many diverging possibilities for motion, all of which would be correct. For this reason, data driven evaluation is not easy to apply to flocking (although a similar method was used successfully by Wang, Ren, Jin, and Manocha (2015)).

In both data driven evaluation and behaviour modelling, the main limitation is access to quality reference data. This issue is especially problematic for a domain such as flocking, this is highlighted in the "FASTSWARM" system (Xiang, Yao, Wang and Jin, 2020). Here a minimization problem is formed, with the most suitable reference velocity from a reference dataset being selected (like the early data driven methods described previously). The authors highlight the difficulty of gathering accurate data on a large swam of insects, citing the limitations of optical sensors as the main issue with lots of tracking errors present even in a controlled lab environment. For the purposes of this project, gathering similar data on flocks of birds is not feasible and as such a data driven approach is not suitable.

References:

Potts, W., 1984. The chorus-line hypothesis of manoeuvre coordination in avian flocks. Nature, 309(5966), pp.344-345.

Girard, M. and Amkraut, S., 1990. Eurhythmy: Concept and process. The Journal of Visualization and Computer Animation, 1(1), pp.15-17.

Reynolds, C., 1987. Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH Computer Graphics, 21(4), pp.25-34.

Reeves, W., 1983. Particle systems—a technique for modeling a class of fuzzy objects. ACM SIGGRAPH Computer Graphics, 17(3), pp.359-375.

2022. [online] Available at: <https://www.red3d.com/cwr/Boids/> [Accessed 22 January 2022].

Warner, B. and Simon, H., 1969. The Sciences of the Artificial. OR, 20(4), p.509.

KARAMOUZAS I., HEIL P., VAN BEEK P., OVERMARS M.: A predictive collision avoidance model for pedestrian simulation. In Proc. 2nd Int. Workshop on Motion in Games (2009), pp. 41–52. 5.

Zanlungo, F., Ikeda, T. and Kanda, T., 2011. Social force model with explicit collision prediction. EPL (Europhysics Letters), 93(6), p.68005.

Karamouzas, I., Skinner, B. and Guy, S., 2014. Universal Power Law Governing Pedestrian Interactions. Physical Review Letters, 113(23).

Paris, S., Pettré, J. and Donikian, S., 2007. Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach. Computer Graphics Forum, 26(3), pp.665-674.

Ondřej, J., Pettré, J., Olivier, A. and Donikian, S., 2010. A synthetic-vision based steering approach for crowd simulation. ACM Transactions on Graphics, 29(4), pp.1-9.

Helmut Lorek and Matthew White., 1993. Parallel Bird Flocking Simulation.

Bo Zhou and Suiping Zhou. 2004. Parallel simulation of group behaviors. In Proceedings of the 36th conference on Winter simulation (WSC '04). Winter Simulation Conference, 364–370.

Reynolds, C., 2006. Big fast crowds on PS3. Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames - Sandbox '06,.

Nilsson, N.J. 1984. Shakey the Robot.

Greggory Hernandez and Curtis Welborn. 2011. Constrained 3D flocking behavior. J. Comput. Sci. Coll. 27, 2 (December 2011), 29–36.

Bayazıt, O., Lien, J. and Amato, N., 2005. Swarming Behavior Using Probabilistic Roadmap Techniques. Swarm Robotics, pp.112-125.

Lerner, A., Chrysanthou, Y. and Lischinski, D., 2007. Crowds by Example. Computer Graphics Forum, 26(3), pp.655-664.

LEE K. H., CHOI M. G., HONG Q., LEE J.: Group behavior from video: A data-driven approach to crowd simulation. In Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (2007), pp. 109– –118.

ALAHI A., GOEL K., RAMANATHAN V., ROBICQUET A., FEI-FEI L., SAVARESE S.: Social LSTM: Human trajectory prediction in crowded spaces. In Proc. IEEE Conf. Computer Vision and Pattern Recognition (2016), pp. 961–971.

LEE J., WON J., LEE J.: Crowd simulation by deep reinforcement learning. In Proc. 11th ACM SIGGRAPH Conf. Motion, Interaction and Games (2018)

Ioannis Karamouzas, Nick Sohre, Ran Hu, and Stephen J. Guy. 2018. Crowd space: a predictive crowd analysis technique. ACM Trans. Graph. 37, 6, Article 186 (December 2018), 14 pages.

GUY S. J., CHHUGANI J., CURTIS S., DUBEY P., LIN M. C., MANOCHA D.: PLEdestrians: a least-effort approach to crowd simulation. In Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (2010), Eurographics Association, pp. 119–128.

Xinjie Wang, Jiaping Ren, Xiaogang Jin, and Dinesh Manocha. 2015. BSwarm: biologically-plausible dynamics model of insect swarms. In Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '15). Association for Computing Machinery, New York, NY, USA, 111–118.

Xiang, W., Yao, X., Wang, H. and Jin, X., 2020. FASTSWARM: A data-driven framework for real-time flying insect swarm simulation. Computer Animation and Virtual Worlds, 31(4-5).