A Reactive Procedural Model of Bird Flight Motion in 3D

Jeff Lee, Breanne Sjodin

CMPUT 414 - Group Small Magellanic Cloud

1 Introduction

Figure 1: Bird flight Sequence example [9]

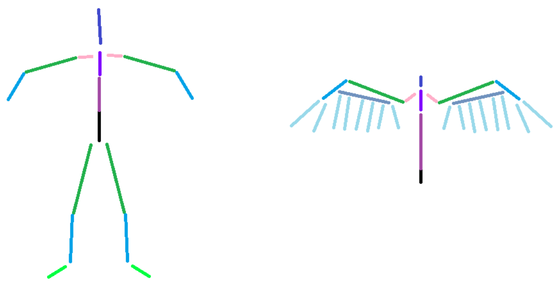

Figure 2: Simplified Comparison of Human 3D rig to Bird 3D rig

Traditionally, animation in computer graphics has relied on interpolating between predefined keyframes made by human animators. Procedural animation is an alternative to this approach where complex motion is generated through a mathematical system from a set of simple parameters. [1] In the context of realtime interactive media such as video games, this can be used to implement reactive animation, i.e. animation which reacts to external inputs such as user control or interaction from other objects. Research shows using reactive animation in games can provide better player experience compared to traditional "canned" animations [13].

Currently, most research on procedural animation for character models has been focused on bipedal and quadrupedal locomotion on terrain. There have been various approaches attempted in this area to generate realistic results. In contrast, procedural animation of avian flight is an essentially unexplored area. This leaves only traditional methods, i.e. keyframe animation or motion capture. The former depends on manual effort from animators, which can be time consuming and labor intensive, and the latter is poorly suited for capturing motion of birds in flight. Ultimately however, both these methods can only produce "pre-baked" results, which are intrinsically ill-suited to apply in realtime interactive scenarios such as video games or VR/AR media. Our model would address these shortcomings by being built from the start with realtime generation and reactivity in mind.

By completing this research we aim to develop a model which can produce aerodynamically plausible flight motion for bird characters based on simple user inputs. Ultimately, we hope this system can be utilized in realtime interactive media such as video games and VR, where its reactive capabilities can shine. In addition, we hope our model can also be used to ease human workloads for non-interactive media such as film and TV.

2 Summary of Existing Work

2.1 Review of Existing Procedural Algorithms

One of the first notable examples of procedural generation is Reynolds' Boids algorithm[17], created in 1986. Boids is a flocking algorithm, developed to have individual object, called a boid, react to other boids with a set of basic rules:

- Cohesion: Each boid flies towards the the other boids.
- Seperation: Each boid also tries to avoid running into the other boids.
- Alignment: Each boid tries to match the vector of the boids around it.

This algorithm produces emergent behavior from a small set of rules, creating data algorithmically rather than manually. Since then, this method of programming has shown potential to solve issues in fields of terrain generation[7], crowd generation[16] and even the in urban planning[10].This technique can be applied to a variety of programming challenges in 2D and 3D animation[1] of characters and environment. Procedural generation of animation in 2D models has been thoroughly explored in innovative ways, one of the most notable being Rain World, [14] an entire game developed around the concept of procedurally generated character movement. The results allowed for unpredictable and fluid animation, creating an immersive world for the player.

The animation produced algorithmically in 3D bipedal movement is hugely beneficial to animation creation processes, allowing for significantly less time and money dedicated to completing projects [18]. When done correctly, this animation is also shown to be comparable in appeal to traditional animation techniques, as demonstrated by a survey comparing bipedal motion capture animation to generated animation[11]. The reactive nature of procedural animation has even been shown to improve player experience in games[13]. Further research into the field of 3D character animation motion in "Fast and Flexible Multilegged Locomotion Using

Learned Centroidal Dynamics" [4] demonstrates a powerful procedure on movement on terrain using methods of a pendulum and centroidal dynamics model for limbs and center of gravity, and novel inverse kinematics for surface interaction.

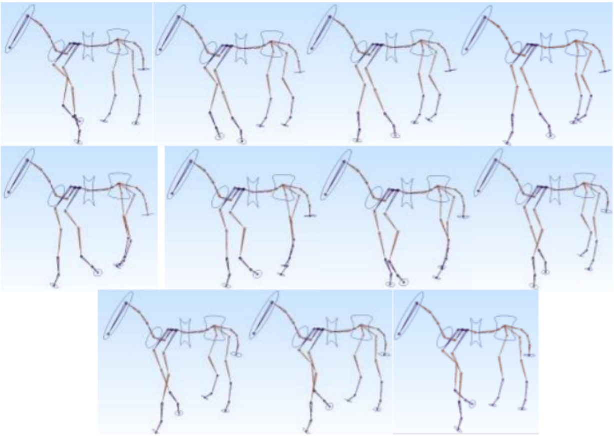

Figure 3: "Phases of walk generated through expressions" from In Expression driven Trigonometric based Procedural Animation of Quadrupeds [8]

In Expression driven Trigonometric based Procedural Animation of Quadrupeds [8], Bhatti et al create fluid animation of a quadrupedal horse, in multiple movement states. They used trigonometric mathematical formula to control each limb movement. This replaces time intensive existing methods of 3D character animation of quadrupeds, such as motion capture with key frames, with a much quicker and more achievable algorithmic approach.

2.2 Review of Previous Bird Animation Models

Getting accurate motion capture data from birds in flight is difficult, and time intensive [15]. In response to this, there have been previous attempts to animate birds algorithmically. A notable example is the physics based method in "Realistic Modeling of Bird Flight Animations" [19], which focused largely on constructing a model that responds to an aerodynamic simulation. The 3D rigged bird model contained feathers that responded directly to drag forces. The feathers are essential to their solution, as the flight patterns of birds are heavily effected by the aerodynamic properties of their feathers. For the controller, they grouped the degrees of freedom for the bird model into a vector, which controlled the birds rotation, translation, and skeletal joints. They split the full motion into a sequence of wing beats, allowing for each beat to respond to the physics effecting the previous. This is similar to what we would like to achieve, though we would like to simplify the process further, ideally creating a comparably believable animation with a model lacking feathers.

$\mathbf{q}_0 \leftarrow \mathbf{p}(0)$
$\dot{\mathbf{q}}_0 \leftarrow \mathbf{v}(0)$
$\mathbf{u}_0 \leftarrow$ null
repeat
  determine $\mathbf{p}'(s)$ and $\mathbf{v}'(s)$ from $\mathbf{q}_0$
  $\mathbf{u}_1 \leftarrow \operatorname{argmin}_{\mathbf{u}} E(\mathbf{S}(\mathbf{q}_0, \dot{\mathbf{q}}_0, \mathbf{F}, \tau_b(\mathbf{u}_0, \mathbf{u}), t), \mathbf{p}'(s), \mathbf{v}'(s))$
  $t \leftarrow u_T - t_b$
  $[\mathbf{q}_0, \dot{\mathbf{q}}_0]^T \leftarrow \mathbf{S}(\mathbf{q}_0, \dot{\mathbf{q}}_0, \mathbf{F}, \tau_b(\mathbf{u}_0, \mathbf{u}_1), t)$
  $\mathbf{u}_0 \leftarrow \mathbf{u}_1$
until $\mathbf{q}_0$ *has reached the end of* $\mathbf{p}(s)$

Figure 4: Entire Flight Synthesis Process from Realistic Modeling of Bird Flight Animations [19]

The algorithm pictured in figure 4, displays the entire process for each wingbeat. Function $u_1()$ maps the degrees of freedom and wingbeat parameter using the Degrees of freedom state, held in the q variable. The torque is determined by the $\tau_b(u_0, u_1)$ function, and F represents the force on the model. All these factors are combined to determine the following state of the bird model. We will be referencing this algorithm to influence our own implementation, and comparing against the results of this method.

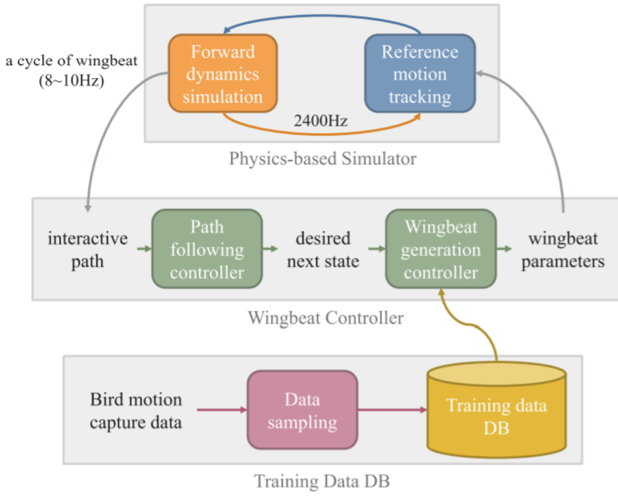

Figure 5: Flight Control System [12]

Data driven motion capture animation of birds in flight was shown to be promising in "Data-Driven Control of Flapping Flight" [12], using data of a dove in flight collected with marker-based optical motion capture and high-speed video cameras. A simplified 3D rig was then created, with the full model containing the rig and flexible feathers. The system algorithm is based structured on a wing beat controller, training data builder, and physics based simulator, as referenced in figure 5.

- Wingbeat Controller: To compensate for limited available training data, and a lengthy amount of degrees of freedom, they implemented a "regression method that can cope with the dimensionality of flapping flight and a new sampling method that achieves dense, well-distributed samples in highdimensional space".[12]

- Training Data: They used State sampling and Wingbeat sampling to generate the training data.

- Physics Simulator: Forward Dynamics driven by internal muscle forces, and external forces of aerodynamics and gravity

The main limit and drawback of motion capture is the difficulty in obtaining accurate data of birds in flight, and the unpredictable effects of small wing movements in aerodynamics. There is also a lack of universality between subjects. It is not the case that we could create a eagle animation using motion capture of a dove, and still achieve accurate results. This issue is present in both physics and motion capture animation methods.

2.3 Review of Fuzzy Control Applications

The concept of fuzzy logic was born from fuzzy set theory. It allows truth values to be real-valued between 0 and 1, in contrast to boolean logic where truth values are binary. Its ability to handle these 'partial truths' has seen many applications in various control systems. In "Enhanced fuzzy finite state machine for human activity modelling and recognition" [5], a fuzzy finite state machine is used to better model human activity. In "Fuzzy control of bipedal running with variable speed and apex height" [3], a fuzzy control scheme is used to determine running motion for a bipedal robot model. This fuzzy controller is called at the apex of each flight phase, and the output controls a lower level state machine which controls running gaits.

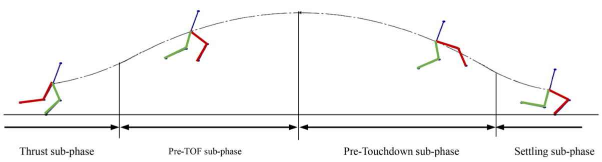

Figure 6: Run Phases for fuzzy controlled biped running [3]

Both these examples show how fuzzy controllers can intuitively model high level control of behavioral and biomechanical activity. This is especially noteworthy in modelling such complex motions as bird flight, which consists of various levels of control hierarchy from trajectory planning to actual wing movement.

3 Implementation

3.1 Challenges

Since our work will be in an area with limited exploration by previous research in procedural animation, we will have to begin by looking at the aerodynamics of real world bird flight. Bird flight is considered to be one of the most complex movements in the animal kingdom, and for the sake of implementation and realtime performance, we will have to reduce and simplify it as much as possible while still maintaining plausibility. Complex aerodynamic factors such as turbulence will also have to be simplified.

3.2 Components

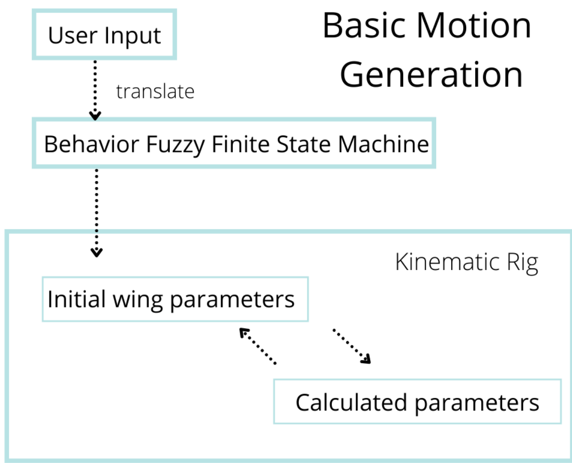

Figure 7: Overview of Control Flow Among Components

Our system is comprised of three components. First, an input stage where the user can define parameters such as acceleration, deceleration, and direction. Secondly, the input stage will pass on these inputs into a fuzzy finite state machine (FFSM) stage, which will translate them into flapping behavior. Third, the resulting motion will be passed onto the kinematic rig, which represents the actual character mesh in 3D. The kinematic rig tracks the motion of the wing-beat and generates the next position of the wing dynamically.

3.3 Rig

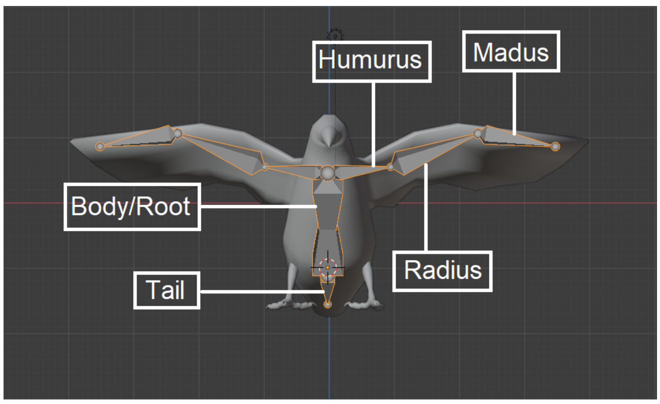

Figure 8: Simplified Bird 3D Rig with Mesh

To create the rig we referenced "Data-Driven Control of Flapping Flight" [12]. We created a simplistic model with 9 total bones and 6 total joints to rotate for flight imitation [2]. Each joint has the ability to rotate freely, each bone is locked on rotation to minimize unexpected behavior. The mesh is skinned as a visual reference exclusively, with deformation as expected behavior. Before manipulating the rig, we placed the skeleton into T-Pose. This allows for simplistic reset of the joint rotations on each iteration of the algorithm [6].

3.4 Controller

We used the up and down arrow keys to shift between high and low frequency values for the wingbeats, simulating acceleration and deceleration. The parameters of Humerus Dihedral/Sweep/Twist, Radius Sweep/Twist, and Madus Sweep angles were controlled by a combination of user input for pitch and and speed.

3.5 Results

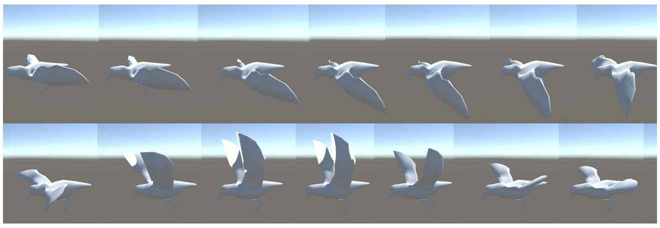

Figure 9: Full Wing-beat Result

Our animation successfully iterates wingbeats procedurally generated by our kinematic functions. Please see un-split full wing-beat in figure 10. User input successfully imitates locomotion of pitch and yaw.

4 Discussion

We have shown that believable motion can be produced algorithmicly with a simplified model lacking in feathers and realistic physics. We hope that this novel simplification will benefit research in procedural generation of flight imitation in the future.

Our model was made with a medium sized bird, with a minimal wingspan in comparison to body size. It would be worth altering the bone lengths to apply this algorithm to various wingspan types. We expect the angles of rotation to require alterations different lengths of wingspan in order to achieve a satisfactory animation.

The mesh we used for a visual reference allowed for deformation, as our goal was to procedurally generate animation of the skeleton. For usage of the algorithm in real world application, a more carefully skinned mesh would be beneficial to remove mesh deformation.

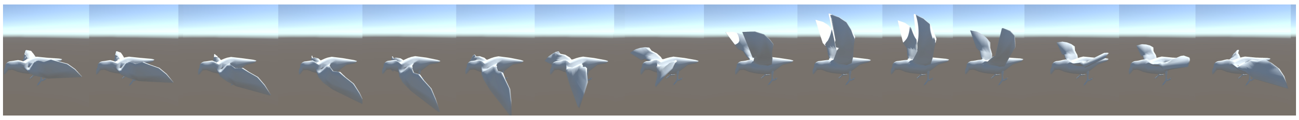

Figure 10: Full Wing-beat Result

References

[1] Diana Phillips Mahoney. *Procedural animation.* 1997. URL: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=rch&AN=9705231610&site=eds-live&scope=site.

[2] Blender Online Community. *Blender - a 3D modelling and rendering package.* Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[3] Behzad Hazrati, Behnam Dadashzadeh, and Maryam Shoaran. "Fuzzy control of bipedal running with variable speed and apex height." In: *International Journal of Dynamics and Control* 7.4 (2019), pp. 1379–1391. ISSN: 2195-268X. URL: https://doi-org.login.ezproxy.library.ualberta.ca/10.1007/s40435-019-00518-9.

[4] Taesoo Kwon, Yoonsang Lee, and Michiel Van De Panne. "Fast and Flexible Multilegged Locomotion Using Learned Centroidal Dynamics". In: *ACM Trans. Graph.* 39.4 (July 2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392432. URL: https://doi-org.login.ezproxy.library.ualberta.ca/10.1145/3386569.3392432.

[5] Gadelhag Mohmed, Ahmad Lotfi, and Amir Pourabdollah. *Enhanced fuzzy finite state machine for human activity modelling and recognition.* 2020. URL: https://login.ezproxy.library.ualberta.ca/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edssjs&AN=edssjs.F3EE0780&site=eds-live&scope=site.

[6] Unity Real-Time Development. *Unity.* Unity Technologies. 2022. URL: https://https://unity.com/.

[7] Santamaría Ibirika et al. *Procedural approach to volumetric terrain generation.* URL: https://link-springer-com.login.ezproxy.library.ualberta.ca/article/10.1007/s00371-013-0909-y. (accessed: 02.07.2022).

[8] Zeeshan Bhatti et al. *Expression driven Trignometric based Procedural Animation of Quadrupeds.* URL: http://arxiv.org/ftp/arxiv/papers/1404/1404.2053.pdf. (accessed: 02.06.2022).

[9] Shuva Brata Deb. *Bird flight sequence.* URL: https://shuva.aminus3.com/image/2008-01-17.html. (accessed: 02.07.2022).

[10] N Maréchal E Guérin E Galin Peytavie. *Procedural Generation of Roads.* URL: https://search-ebscohost-com.login.ezproxy.library.ualberta.ca/login.aspx?direct=true&db=a9h&AN=50891866&site=ehost-live&scope=site. (accessed: 02.07.2022).

[11] Prashant Goswami Eric Ahlström Lucas Holmqvist. *Comparing Traditional Key Frame and Hybrid Animation.* URL: https://dl-acm-org.login.ezproxy.library.ualberta.ca/doi/pdf/10.1145/3099564.3106640. (accessed: 02.06.2022).

[12] JUNGDAM WON EUNJUNG JU and JEHEE LEE´. *Data-Driven Control of Flapping Flight.* URL: https://dl-acm-org.login.ezproxy.library.ualberta.ca/doi/pdf/10.1145/2516971.2516976. (accessed: 02.15.2022).

[13] Chelsea Hash and Katherine Isbister. *Reactive animation and gameplay experience.* URL: https://dl-acm-org.login.ezproxy.library.ualberta.ca/doi/10.1145/2159365.2159428. (accessed: 02.06.2022).

[14] James Therrien Joar Jakobsson. *Animation in Rain World.* URL: https://www.youtube.com/watch?v=sVntwsrjNe4&t=1072s. (accessed: 02.06.2022).

[15] Andrew A. Biewener2 Marc E. Deetjen1 and David Lentink1. *High-speed surface reconstruction of a flying bird using structured light.* URL: https://journals.biologists.com/jeb/article/220/11/1956/17940/High-speed-surface-reconstruction-of-a-flying-bird. (accessed: 02.06.2022).

[16] Nuria Pelechano Otger Rogla Gustav A.Patow. *Procedural crowd generation for semantically augmented virtual cities.* URL: https://www-sciencedirect-com.login.ezproxy.library.ualberta.ca/science/article/pii/S0097849321001291. (accessed: 02.07.2022).

[17] Craig Reynolds. *Flocks, herds and schools: A distributed behavioral model.* URL: https://dl-acm-org.login.ezproxy.library.ualberta.ca/doi/abs/10.1145/37401.37406. (accessed: 02.06.2022).

[18] Timothy E. Roden. *Procedural content creation and technologies for 3D graphics applications and games.* URL: https://digital.library.unt.edu/ark:/67531/metadc4726/. (accessed: 02.06.2022).

[19] Jia-chi Wu and Zoran Popovic´. *Realistic Modeling of Bird Flight Animations.* URL: https://dl-acm-org.login.ezproxy.library.ualberta.ca/doi/pdf/10.1145/882262.882360. (accessed: 02.15.2022).