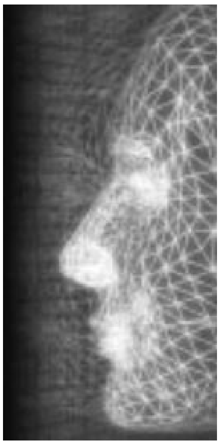
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2006; **17**: 199–206
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cav.123

Autonomous Boids



By Christopher Hartman and Bedřich Beneš[*]

The classical work of bird-like objects of Reynolds simulates polarized motion of groups of oriented particles, bird-like objects, or simply boids. To do this, three steering vectors are introduced. Cohesion is the tendency of boids to stay in the center of the flock, alignment smoothes their velocities to similar values, and separation helps them to avoid mutual collisions. If no impetus is introduced the boids wander somewhat randomly so an external leading force is necessary for the correct flock behavior. As can be observed during the bird flocking in the fall, birds sometimes move in a way that is not captured by the above described framework. Some of the birds, typically the ones on the edge of the flock, suddenly shoot-off. The flock then pursues this leader. In the original work by Reynolds the cohesion and separation are two complementary steers. We introduce a complementary force to the alignment that we call the change of leadership. This steer defines the chance of the boid to become a leader and try to escape. The leadership is derived from the boid position and the flock eccentricity. If a boid is on the front edge of the flock it has a higher chance to escape. Escaping from the flock is simulated as a sequence of velocity increases that are added to the current velocity of the boid. The entire system is easy to implement, is efficient, and runs simulations of hundreds of boids on a standard computer at 30 frames per second. Our system is aimed to real-time simulations and has the potential to be used in games, crowd simulations, etc. Copyright © 2006 John Wiley & Sons, Ltd.

Received: 10 April 2006; Revised: 2 May 2006; Accepted: 10 May 2006

KEY WORDS: flock; boid; computer animation; artificial life; procedural animation; particle system

Introduction: Boids and Artificial Life

The classical paper *Flocks, herds and schools: a distributed behavioral model* of Reynolds[1] was cited so many times and extended in so many different ways (see the next section for more details) that one could think that there is almost nothing that could be added. Many of the extensions introduce additional properties of the bird-like objects (*boids*), some describe constrained solutions, some extend the previous work in spite of computational complexity, some tend to easy solutions usable in computer games, etc.

An interesting look at boids can be taken from the perspective of artificial life. This discipline has been clearly defined over years and one of its fundamental rules states that the holistic emergent phenomena is the result of interactions of independent entities.[2] Each entity has local rules for behavior yet there are no global rules at all. This condition, however, is broken by the classical flock simulations. There is always a global direction, leader with a specified behavior, or a global force that determines the flock's motion. Boids behave chaotically when there is no global direction, force, or leader. They either find a certain configuration that is optimal for the given set of parameters, they move randomly if some random motion is introduced, they oscillate, or they form regular patterns.

As can be observed during bird flocking in the fall (see the videos on the web page,[3]) if there is no specific external condition birds do not behave in this way. Some species of birds 'play a game' where they fly and follow a leader bird from the flock. A flock of birds can play this game for hours. A bird, typically one on the edge of the flock, suddenly shoots-off outside the flock. The other birds follow. After a period of time, this bird flies back into the flock. Then, another bird tries to escape, and the game goes on. The leadership is not tied to a single bird. It can be observed that this role is dynamic

[*]Correspondence to: Bedřich Beneš, Department of Computer Graphics Technology, Purdue University, Knoy Hall of Technology, 401. N.Grant Street West Lafayette, IN 47907-2021, USA. E-mail: bbenes@purdue.edu

and changes over time. A flock of birds can suddenly change the direction when this role is taken on by another bird.

Based on this observations we propose an extension of the originally described idea of Reynolds.[1] The three flock-forming steer forces (cohesion, alignment, and flocking) are extended by *the change of leadership*. In Reynold's original research flocking and the cohesion can be thought of as the two complementary steers. In our research, the change of leadership, can act as a complementary force to the alignment. It is important to note that by this extension the Reynold's system is not changed. Our system is a superset of his original concept. By activating the leadership steer, the flock behaves realistically. The emergent phenomenon, the flock formation, pops up without a need for an external influence. In other words, the system is described as purely artificial life.

Previous Work

Reynolds[1] describes the flock behavior as a result of the motion and the interaction of boids. Each boid has the three local rules of behavior.

- Cohesion (flocking) is the tendency of a boid to stay close to the center of the local flock formed by its neighbors.
- Separation (collision avoidance) is the tendency of boids to stay far from its neighbors and to avoid the potential collisions. The cohesion and the separation are somewhat complementary, but are not the same force with the opposite sign.
- Alignment (velocity matching) is the tendency of a boid to match the direction and the speed of its neighbors. This is the most important factor causing boids to follow each other.

Our research is an extension of Reynold's[1] original work. This paper describes these steers and the way in which they are calculated in detail in the next section.

One of the extensions of the previously mentioned paper recently introduced the same author in Reference [4]. Autonomous character are navigated in a virtual environment. Their motion is described at the three different levels. At the lowest level individual kinds of behavior, such as seek, flee, pursuit, follow the flow, avoid an obstacle, are described. At the middle and the top levels, these behaviors are combined together.

Tu and Terzopoulos[5] introduced a logical extension to boids. Their autonomous agents, fishes, are equipped with a perceptual system, a simplified neural network-based brain, an intention generator, and behavior routines that control motor controllers. Each fish behavior is driven by the tree steers; hunger, libido, and fear. The response of the fish is defined by its internal state and external conditions. The resulting response controls the muscles that, together with the environment response, result in the motion of the fish.

Musse *et al.*[6] simulated a crowd behavior in a collaborative virtual environment. A crowd is modeled as an artificial life system, where every agent has its local rules of behavior and the emergent phenomena is the crowd behavior that results from a seeing certain goals.

Anderson *et al.*[7] introduced a constrained group of animations. Their flocks of agents move on a predefined path. Whereas the traditional constrain-based animation is either path-following or key-framed, they use the hard constraints by sampling the space of possible cases and evaluate the most appropriate one. Results of their work are aimed for a generation of off-line animations.

Lai *et al.*[8] described another constraint-based system for flock/crowd behavior using group motion graphs. The alternated trajectories are generated by modifying the existing data that is stored in the group motion graph. The graph is generated prior to the simulation and the data is exploited during the motion generation.

Ryder *et al.*[9] used controlled flock behavior to increase realism of scenes with a cultural heritage environment. Their crowds naturally wander through the scene. Their virtual characters go from place to place, and eventually meet and join. Ryder *et al.* also discuss in their paper rendering details such as levels of detail techniques.

Ulicny *et al.*[10] show another approach to crowd modeling. Their interactive application uses brush-like tools for crowd distribution. They simulate crowd distribution in a theater and a pedestrian crowd in a virtual city.

A very good starting point for future research in boids, flocks, herds, swarms, etc. is the web page of Reynolds[3] which discusses boids from different perspectives. It provides useful links to the relevant resources on Internet and to free software packages (such as Open Steer[11]) that implement, or to some extent improve, the concept of boids.

The next section of this paper continues with a description of boids and how the addition of the fourth steer, a change of leadership, relates to artificial life and autonomous behavior. The paper then continues with a discussion of the animation and implementation aspects of the new technique. The paper concludes section describing results, conclusions, and some open questions for future work.

Boids

In this section we will describe the implementation of boids that is based on Reynold's original paper of Reference [1]. The entire flock is made up of the set B of n boids b_i,

$$B = \{b_i, i = 0, 1, \ldots, n-1\}.$$

Each boid has a set of the following attributes: position p_i, velocity $\vec{v}_i$, the up-vector $\vec{u}_i$, and the three steering forces (separation, cohesion, and alignment). The forces are described in detail in the following text.

An important aspect in the algorithm is the boid's vision. Boids have limited vision that is defined by a variable that corresponds approximately to the five times the boid's size. Every boid has associated a visibility sphere and everything that is inside is visible by the boid. This variable can be set by the user and it is a global parameter that is equal for every boid in the flock and it is denoted by e. Let us denote the set of boids inside the sphere of visibility of the $i-$th boid by V_i. The entire visible set is then described as:

$$V_i = \{b_j \in B;\ \forall b_j :\ |b_i - b_j| < e, j = 0, 1, \ldots, m-1\},$$

where m is the number of the boids visible by the ith boid. In next section the three steers are described.

Separation

Every member of a crowd tends to avoid collision with its neighbors. This tendency is called separation or collision avoidance. It steers a boid to avoid overcrowding the local flockmates. If only this steering force is applied the flock will dissipate.

There are many ways how this force can be implemented. A simple but efficient solution is depicted in Figure 1. Vectors defined by the position of the boid b_i and each visible boid b_j are summed and the separation steer, denoted by $\vec{s}_i$, is calculated as the negative sum of these vectors:

$$\vec{s}_i = -\sum_{\forall b_j \in V_i} \left(p_i - p_j\right). \qquad (1)$$

Cohesion

Cohesion, or flock centering, is the steering force which moves a boid toward the center of the visible flock. It acts as the complement to the separation (see in Figure 2). If only this steer is applied, the flockmates will merge into one single position. Cohesion of the boid b_i is denoted by $\vec{k}_i$ and is calculated in two steps.

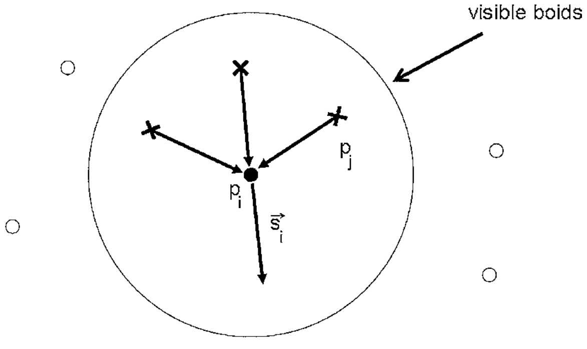


Figure 1. Separation $\vec{s}_i$ is expressed as the sum of all vector corresponding to the oriented lines p_i, p_j. The visible boids are denoted by the cross. The boids not directly visible by the b_i are depicted by the empty circles. The boid b_i is described by the full circle and its position is denoted by p_i.

First, the center of the directly visible set V_i is calculated. This center is denoted by c_i and corresponds to the center of density of all visible boids:

$$c_i = \sum_{\forall b_j \in V_i} \frac{p_j}{m}. \qquad (2)$$

The tendency of the boid b_i to navigate toward the center of density of the visible flock V_i is calculated as the cohesion displacement vector $\vec{k}_i$:

$$\vec{k}_i = c_i - p_i \qquad (3)$$

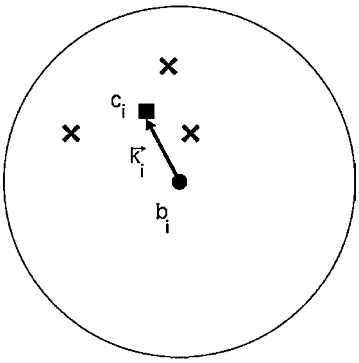


Figure 2. Cohesion $\vec{k}_i$ is calculated as the vector pointing to the center c_i (denoted by the black square) of the flock of visible boids.

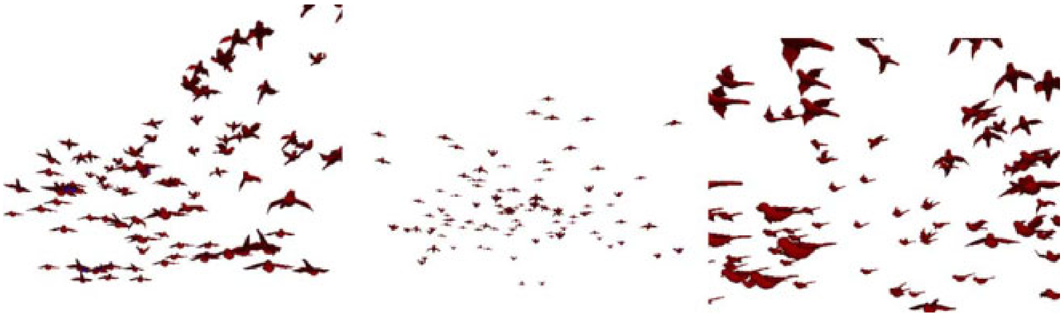


Figure 3. Example of extreme flocking (left), high separation of boids (middle), and alignment (right).

There is a special case when no boid is around the b_i and $m = 0$. In this case the Equation (2) is not defined; no cohesion is applied; and, its result is the zero vector $\vec{k}_i = \vec{o}$.

Alignment

The last steering force, alignment, or velocity matching, is important for the boids to follow an impetus. Boids tend to align with the velocity of their flockmates. This steer is denoted by $\vec{m}_i$ and is calculated as the average velocity of the visible flockmates

$$\vec{m}_i = \sum_{\forall b_j \in V_i} \frac{\vec{v}_j}{m}. \qquad (4)$$

The velocity speed is the size of the vector. Therefore, the boids will automatically slow down or speed up depending on their flockmates. If a boid accelerates too much it can jump out of the visibility sphere of the flockmates and eventually escape.

If there is no boid directly visible, the cardinality of V_i is equal to zero and $\vec{m}_i = \vec{o}$.

Combining the Steers

All the steers are combined into one resulting influence. Let us recall that the vectors are separation $\vec{s}_i$, cohesion $\vec{k}_i$, and alignment $\vec{m}_i$. The new position p'_i of the boid in the time $t + \Delta t$ is calculated from its velocity $\vec{v}_i$ and the previous position as

$$p'_i = p_i + \Delta t \vec{v}_i.$$

To reflect the steering forces the velocity is first updated correspondingly

$$\vec{v}'_i = \vec{v}_i + S.\vec{s}_i + K.\vec{k}_i + M.\vec{m}_i,$$

where S, K, and M are coefficients describing influence of each force. These coefficients are set by the user and are global to the flock.

Figure 3 shows examples of flocking based on different parameters.

Swarm behavior can be obtain by setting the velocity matching (alignment) to zero.

The values of the parameters S, K, and M should be within the range $\langle 0, 1 \rangle$. The key factor for correct flock behavior is its initializations. The initial velocity should be 'reasonable' as well as the flock initial distribution. Ideally each boid should be visible by at least one boid. In this way each boid will indirectly influence each other and the flock will not break into parts. Also, if the initial speed is too high, the flock can eventually break into parts. We have found that setting the initial velocity to zero and letting the simulation to run for a while helps the system to find a stable configuration.

Leadership

The new contribution to the flocking algorithm is the change of leadership. As mentioned above, boids without an external force such as a leader, a global aim, or a direction of the flight, will find optimal configuration in the three-dimensional space and stay still, oscillate, or form obvious visual patterns.

This does not correspond to observations of real flocks. Birds 'play a game' in which a random leaders shoot-off from the flock and the others bird pursuit him. The flock follows the runaway and then absorbs him back into the flock.

The change in leadership is evaluated across the visibility sphere of each boid. As a result, only a local flockmates are considered. A boid inside a dense flock will not take the leadership, because it cannot see a clear space to fly. Boids on the edge of the crowd will have higher chance to escape into the boid-free space. Also a boid that is on the edge of the flock but is behind it, will not fly away. This is because it involves too great of a change in direction. We can conclude that only boids on the border flying in the direction of the flock or being very slow will have the tendency to escape and are good candidates to become a leader of the flock.

Eccentricity

To determine possible candidates, we evaluate if the boid is inside or outside the flock. To express this we first define a measure that states the composition of the visible flockmates V_i. We define so called eccentricity of the flock and denote it by x_i. The eccentricity is a scalar value and is calculated as the distance of the boid b_i from the center of the density c_i (see Equation (2)) of the flock:

$$x_i = \sum_{\forall b_j \in V_i} \frac{p_j}{m.e} = \frac{c_i}{e}. \qquad (5)$$

The divisor e is the visibility range of the boid, so the value x_i is normalized $x_i \in \langle 0, 1 \rangle$. Value $x_i = 0$ identifies a boid equally surrounded by the flockmates, values close to one identify a boid on the edge of the local flock. In the special case of an abandoned boid $m_i = 0$ and we set explicitly $x_i = 0$.

There is also a global parameter of the influence of the eccentricity to the boids. This parameter has the same function as the K, S, and M and is denoted by X.

Velocity and the Boid's Position

The value x_i is the normalized size of the cohesion k_i (see Equation (3)). Comparing the direction of the cohesion with the direction of the velocity of the boid, we can detect whether the boid is behind or in front of the flock. Recall that the cohesion points to the center of the flock. By calculating the dot product of the normalized vectors $\vec{v}_i$ and $\vec{k}_i$ we determine the orientation of the boid with respect to the flock as can be seen in Figure 4.

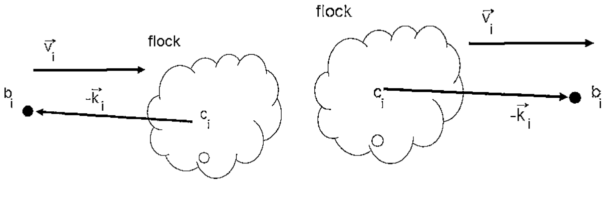


Figure 4. Explanation of the leadership calculation. Chance to shoot-off is high for boids in front of the flock.

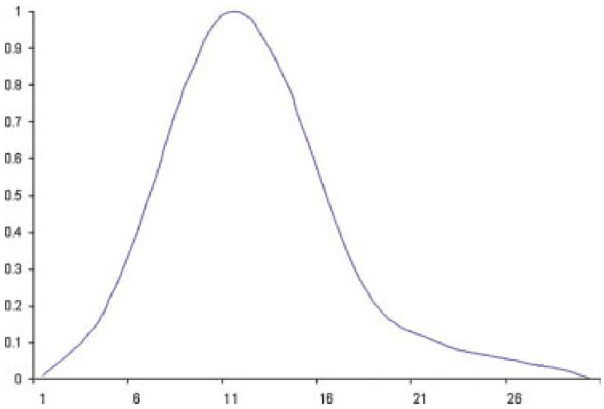


Figure 5. Velocity changes over the actual frame as the boid takes the lead.

If the dot product

$$-\frac{\vec{k}_i}{|\vec{k}_i|}.\frac{\vec{v}_i}{|\vec{v}_i|} < 0$$

the boid is behind the flock, if the value is positive it is in its front. The closer the value is to one, the more in front the boid is. The closer the value is to minus one, the more behind the flock the boid is.

To avoid reentrance, each boid is assigned a status that determines if it is actually escaping or not. In other words the boid that takes the leadership cannot take it again while being a leader.

Also, it is difficult to express the process of running away the flock purely by vector algebra. For this reason, we use the above described steps as a trigger to decide whether the boid escapes from the flock or not. Once it is determined, its velocity is increased according to the function its shape is in Figure 5. This function describes the speed increment over the time. Entire escape is about 3 seconds long. This function is correlated with the actual velocity of the escaping boid.

Algorithm

The entire process of the change of the leadership is described by the following steps:

1. If the boid's status is 'a leader' do nothing.
2. Calculate the cohesion $\vec{k}_i$ (Equation (3)).
3. Calculate the eccentricity x_i (Equation (5)).
4. if ($x_i < X$*`GaussRand01()`) do nothing.
5. Set the boid's status to 'a leader'.

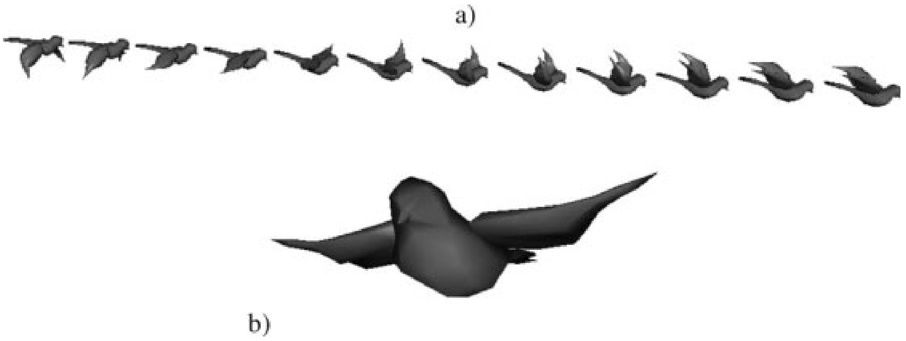


Figure 6. a: The bird was modeled and animated in Maya. The models were exported and twelve frames of the animation were used. b: A descending bird.

6. Apply the function from the Figure 5 for the next few seconds.
7. Set the boid's status to 'not a leader'.

The first line avoids the reentrance. The lines two and three are the application of the corresponding formulas. The fourth line takes the global parameter of the influence of the cohesion X and multiplies the actual result by x_i. This is scaled down by the normalized Gaussian random number `GaussRand01()` with $\mu = 1/2$. If this condition is true, the process of runaway is triggered (line five) and the boid starts escaping. When this happens, the boid's velocity is successively increased according to the predefined function.

The algorithm uses vector calculations and one random number generator call. The random numbers are actually stored in a look-up table to speed-up the calculation.

The calculated velocity vector is just added to the actual one as in the case of alignment, separation, and cohesion.

Implementation and Results

The application is written in C++ and uses OpenGL, GLUT, and GLUI. The system uses vertex buffers for the model animation and runs scenes up to 250 boids effectively (about 30 fps) on an average laptop (1.6 GHz, 1 GB RAM, Quadro FX Go 1440). The entire application is aimed to real-time simulations and the actual bottleneck is the calculation of steers, where each boid is compared with each for visibility. We construct the visibility matrix in each frame which stores the distance between each pair of boids. This is also calculated with $\mathcal{O}(n^2)$, but it is constructed just once per frame, and in fact only one half needs to be calculated, since it is symmetrical. This is used as the look-up table with linear access. Different acceleration techniques could be implemented.

The bird model was created and animated in MAYA and exported to our system. Twelve frames from the animation are displayed in Figure 6. When the boids are initialized, each has assigned a random local counter of frames that is used as the Index of which model is actually rendered. In this way, the probability of two boids being rendered with the same instance is small.

Each bird can be in one of two modes, flying straight or up, or descending. If the bird is descending its animation is run till the wings are completely open as can be seen in Figure 6b. This object is then rendered repeatedly while the bird is flying down. At the moment a bird needs to do some effort to fly we switch its status to active and the animation continues by exchanging the displayed models as shown in Figure 6a.

The flock of birds is initialize randomly. When the leadership is not introduced it behaves exactly as the Reynold's[1] original model. When the value of leadership is increased boids start to shoot-off and are followed by the others as soon as they slow down and are captured by the flock. Higher values of leadership eventually destroy the flock and break it into small parts. Because of the maximal speed, one boid abandons the flock very rarely.

The trajectory of the boid can be stored and displayed as the boid's trace as can be seen in Figure 7. We have found this visualization to be really practical for understanding the flock dynamics. A boid can be displayed as an animated model or as a simplified model. In the simplified model a boid is represented as sphere with a line that shows the heading. Snapshots

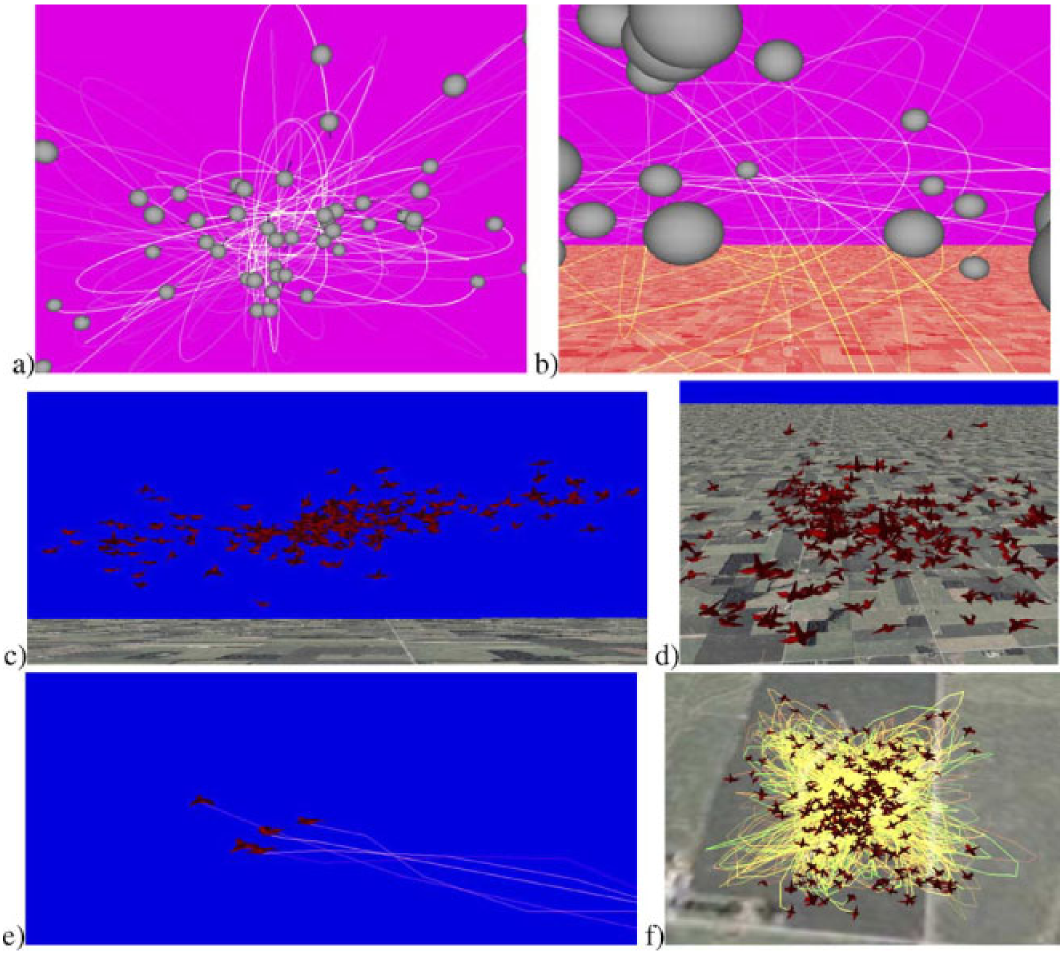


Figure 7. (a) Boids are simplified by spheres with the heading vector and their traces are showed. The image shows the outside view and the image (b) the boid's view. The image (c) is the outside view and the image (d) is the boid's view from the simulation of the change of the leadership. The image (e) shows a small flock of boids with a change in leadership and the image (f) shows a trace of a flock of 250 birds.

from the application are in Figure 7. The left one shows the outside view, whereas the right one shows the boid's view.

We believe that our model provides more realistic results than the original one. This is a difficult claim to prove since we relay solely on visual comparison and is generally true in any research in computer animation. To this extent our model is logically correct in that sense that it describes the real behavior of birds as observed. It implements this behavior in a straightforward and easy-to-understand and easy-to-follow way and it provides results that are expected. A correct verification of the behavior should involve some statistical analysis, but we do not know any that has been done with the flocks of real birds.

Conclusions

We have shown that a simple extension of the idea of boids by the change of the leadership leads to another degree of realistic flocking. Our application is a clear extension of Reynold's original work that is its special case when the leadership is set to zero. The major advantage of our technique is that we do not need to define a global force or global leader. This is seamlessly integrated into the flocking model. The user has a parameter that is the boid's leadership attribute. This defines the chance of becoming leader if the boid is on the border of the flock. If this condition is satisfied, the boid eventually shoots-off the flock and the others, because of cohesion and velocity matching, will follow it.

Our models provides realistic results that are in tune with the expectations and correspond visually to the flock behavior.

The model is aimed to real-time graphics, is fast, and is a straightforward extension of the original boid models. We have intentionally developed an easy-to implement model and the model was implemented and tested by an undergraduate student.

Future work should focus on testing the model in different situations such as cultural heritage[9] or constrained crowds.[7,8] We would like to implement the entire system in a Virtual Reality hardware such as FLEX or CAVE to really experience the flight of boids. Another future work also includes realistic bird animation, for example such as described by Wu and Popović.[12]

ACKNOWLEDGEMENTS

We would like to express our gratitude to Jeff Ballinger for creating the bird models, animations, and textures. This work was done with the support of the Envision Center for Data Perceptualization of the Purdue University.

References

1. Reynolds CW. Flocks, herds and schools: a distributed behavioral model. In *SIGGRAPH'87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press: New York, 1987; 25–34.
2. Liekens A. *Artificial Life*. Macmillan Press Ltd.: Basingstoke, UK, 2002.
3. Reynolds CW. A Web Page Related to Boids, Flocks, and Swarms. http://www.red3d.com/cwr/boids/.
4. Go J, Vu T, Kuffner JJ. Autonomous behaviors for interactive vehicle animations. In *SCA'04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM Press: New York, 2004; 9–18.
5. Tu X, Terzopoulos D. Artificial fishes: physics, locomotion, perception, behavior. In *SIGGRAPH'94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* ACM Press: New York, 1994; 43–50.
6. Musse SR, Babski C, Capin T, Thalmann D. Crowd modelling in collaborative virtual environments. In *VRST'98: Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM Press: New York, 1998; 115–123.
7. Anderson M, McDaniel E, Chenney S. Constrained animation of flocks. In *SCA'03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association: Aire-la-Ville, Switzerland, 2003; 286–297.
8. Lai Y-C, Chenney S, Fan S. Group motion graphs. In *SCA'05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM Press: New York, 2005; 281–290.
9. Ryder G, Flack PA, Day AM. Adaptive crowd behaviour to aid real-time rendering of a cultural heritage environment. In *VAST*, Fellner DW, Spencer SN (eds). Eurographics Association, 2004; 29–36.
10. Ulicny B, de Heras Ciechomski P, Thalmann D. Crowdbrush: interactive authoring of real-time crowd scenes. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, Pai DK, Boulic R (eds). Eurographics Association: Grenoble, France, 2004; 243–252.
11. Open Steer. Steering Behaviors for Autonomous Characters. http://opensteer.sourceforge.net/.
12. Wu JC, Popović Z. Realistic modeling of bird flight animations. *ACM Transactions on Graphics* 2003; **22**(3): 888–895.

Authors' biographies:

Christopher Hartman is undergraduate student at the Department of Computer Graphics Technology at the Purdue University. He is interested in real-time animations and programming.

Bedrich Benes received his PhD in Compute Graphics from the Czech Technical University in 1998. Since August 2005 he is an Assistant Professor at the Department of Computer Graphics Technology at the Purdue University. He is coauthor of three books about computer graphics and more than twenty papers. His research interests include procedural modeling, artificial life, realtime rendering, and global illumination.