CHAPTER

11 Behavioral Animation

As discussed in the previous chapters, there are a variety of ways to specify motion. Some are very direct. In key-frame animations, for example, the animator precisely specifies values for animation parameters as specific frames and selects an interpolation procedure to fill in the remaining values for the in-between frames. There is not much mystery as to what the parametric values will be or what the motion will look like. In physically based animation, such as rigid body dynamics, the animator might specify starting values for some of physical simulation parameters, for example, initial position and velocity. The animator generally knows what the animation will look like but does not have a good idea of the intermediate parametric values as an object falls, collides, and bounces. This is an example of model-based animation where there is an underlying model controlling the motion. In this case the model is physics—or at least some approximation to it.

Behavioral animation is another type of model-based animation in which it is the cognitive processes that are being modeled. The term *cognitive process* refers to any behavior that can be attributed to mental activity, voluntary or involuntary, that is reacting to the environment—basically non-physically based procedural motion. The cognitive processes can be as simple as moving toward a light or as complex as balancing self-preservation, mood swings, and personality disorders.

Whereas modeling complex cognitive processes might be necessary for creating a believable autonomous character in a game, simple cognitive processes are often sufficient for modeling crowd behavior. Development of behavioral animation has been motivated by applications of virtual reality (VR), computer games, military simulations, and the film industry. Behavioral animation can be used to evaluate environmental design such as escape facilities in a burning building. It also can be used to create massive crowds for films, to supply background animation, or to provide stimulus for virtual environments.

Modeling behavior can be useful for several reasons:

- To relieve the animator from dealing with details when background animation is not of primary importance
- To generate animation on the fly, as in a game or VR application
- To inspect the results of a simulation, that is, to evaluate the environment relative to a certain behavior
- To populate virtual environments

Synthetic characters with built-in rules of behavior are often referred to as *actors* or *intelligence agents*. When a figure is meant to be the embodiment of a user in a virtual environment, the term *avatar*[1] is often used.

[1]In Hindu mythology, an *avatar* is an incarnation of a deity (especially Vishnu) in human or animal form.

At the highest levels of abstraction, the animator takes on the role of a director. At his command are intelligent characters who know how to get the job done, they only need to be told what to do. It is the character's job to find a way to do it. The animator ceases to animate and only directs the action in a very general way. This chapter is about behavior and how it is modeled in computer animation applications. The implied objective is that the behavior be realistic, or at least believable, so that character appears to be *autonomous*.

There are two general aspects of behavioral animation: cognitive modeling and aggregate behavior. Cognitive modeling can range from a simple cause-and-effect-based system to an attempt to model psychological changes in an agent's mental make-up under environmental influences. This aspect of computer animation is in the realm of artificial intelligence (AI). Aggregate behavior refers to how the individuals making up a group are modeled and how their behavior contributes to the quality of the group's motion as a whole.

Cognitive modeling

In general, the task of modeling intelligence and cognition is the problem domain of AI. Aspects of behavior modeling are addressed in computer animation because of the spatial aspects and the imagery produced, and not by focusing on the accuracy of the mental processes being modeled. At all levels of cognitive modeling, the character senses the environment, processes the sensory information possibly with respect to his/her internal state and traits, and then responds with actions that interact with the synthetic environment and possibly with changes to the internal state. At the basic level, a simple reasoning mechanism, such as use of a decision tree or case-based reasoning, suffices. More sophisticated approaches might model personality and mood of the character. For example, a rule-based approach is often used and can prove effective, although there are problems of scale. A common approach is to use a database of simple actions (e.g., simple SIMS games) as a library of behaviors and have appropriate behavior kick in when conditions are satisfied. Simple rules of the form (<cond>,<behavior>) can determine activity with the implication of executing the activity when the condition is met: if (<cond_i>is TRUE) then do activity_i. Of course, the condition can be arbitrarily complex as can the activity.

When more than one precondition is met, some arbitration strategy must be in place. A simple method is to say that the first condition satisfied is activated, so order of rules sets the priority of rules. For a simple example, consider the behaviors of walking, running, standing, and turning. One of the major issues with this approach is transitioning from one activity to another. The simple solution is to have a neutral behavior to start and end each behavior, for example, standing.

$$\text{stand} \rightarrow \text{walk} \rightarrow \text{turn} \rightarrow \text{run} \rightarrow \text{stand}$$

A more sophisticated approach is to blend from one activity to another with either precomputed transitions or transitions computed on the fly. Adding more activities requires that more rules are added and more transitions are needed. Meta-rules can be added to reason about the rules such as when to apply, which are compatible, and so forth.

Higher levels of cognitive modeling often refer to themselves as modeling intelligence and individuality. Instead of directly reacting to environmental conditions, a cognitive structure is created that can more realistically sense the environment, alter the mental state, and reason about conditions using internal representations while filtering everything through mental states.

Aggregate behavior

Managing complexity is one of the most important uses of the computer in animation, and nothing exemplifies that better than particle systems. A *particle system* is a large collection of individual elements, which, taken together, represent a conglomerate fuzzy object. Both the behavior and the appearance of each individual particle are very simple. The individual particles typically behave according to simple physical principles with respect to their environment but not with respect to other particles of the system. When viewed together, the particles create the impression of a single, dynamic complex object. This illusion of a greater whole is referred to as *emergent behavior* and is an identifying characteristic of particle systems, flocking, and, to a lesser extent, crowds.

Often behavioral animation is concerned with a large number of characters. The primitive behaviors of flocking and prey–predator activity, as well as crowd modeling, are common examples. One of the problems when dealing with a large group of behavioral characters is that knowledge about nearby characters is often required. Particle systems do not exhibit this complexity because there is (typically) no particle–particle interaction. There is only simple physical interaction with the environment. But when knowledge of nearby characters is required, the processing complexity is n-squared where n is the number of characters. Even when interactions are limited to some k nearest neighbors, it is still necessary to find those k nearest neighbors out of the total population of n.

One way to find the nearest neighbors efficiently is to perform a three-dimensional bucket sort and then check adjacent buckets for neighbors. Such a bucket sort can be updated incrementally by adjusting bucket positions of any members that deviate too much from the bucket center as the buckets are transformed along with the flock. There is, of course, a time-space trade-off involved in bucket size—the smaller the buckets, the more buckets needed but the fewer members per bucket on average. This does not completely eliminate the n-squared problem because of worst-case distributions, but it is effective in practice.

The members of a flock, typically fewer in number than particles in a particle system, usually behave according to more sophisticated physics (e.g., flight) and a bit of intelligence (e.g., collision avoidance). Simple cognitive processes that control the movement of a member are modeled and might include such behavior as goal-directed motion and the steering to maintain separation from neighbors.

Adding more intelligence to the members in a group results in more interesting individual behaviors, which is sometimes referred to as *autonomous behavior*. Modeling autonomous behavior tends to involve fewer participants, less physics, and more intelligence. Particle systems, flocking, and crowd behavior are examples of independently behaving members of groups with varying levels of autonomy, physical characteristics, and simulated motions (Table 11.1). If a larger number of autonomous agents are modeled, crowds are created that share some of the same emergent qualities as flocks and particle systems.

Table 11.1 Aggregate Behavior

Type	Number of Elements	Incorporated Physics	Intelligence
Particles	10^2–10^4	Much—with environment	None
Flocks	10^1–10^3	Some—with environment and other elements	Limited
Crowds	10^1–10^2	Usually little physics, but depends on interaction with environment	Varies from little to much

11.1 Primitive behaviors

Primitive behavior, for purposes of this discussion, is characterized by being immediately reactive to the sensed environment as opposed to any temporally extended reasoning process. The environment is sensed and an immediate reactive behavior is made to the conditions extracted from the sensory data. There is little reasoning, no memory, and no planning. Usually, a simple rule-based system is involved. While a limited number of internal states might be modeled, they control basic urges such as hunger and flight-from-danger. Flocking and prey–predator behavior are the primary examples of such primitive behavior that is of a direct cause-and-effect form.

11.1.1 Flocking behavior

Flocking can be characterized as having a moderate number of members (relative to particle systems and autonomous behavior), each of which is controlled by a relatively simple set of rules that operate locally. The members exhibit limited intelligence and are governed by basic physics. The physics-based modeling typically includes collision response, gravity, and drag. As compared to particle systems, there are fewer elements and some interactions with nearby neighbors are modeled. In addition, members' behavior usually models some limited intelligence as opposed to being strictly physics based.

Flocking is one of the lowest forms of behavioral modeling. Members only have the most primitive intelligence that tells them how to be a member of a flock. From these local rules of the individual members, a global flocking behavior can emerge. While flocks typically consist of uniformly modeled members, prey–predator behavior can result from mixing two competing types of mobile agents.

The flocking behavior manifests itself as a goal-directed body, able to split into sections and re-form, creating organized patterns of flock members that can perform coordinated maneuvers. For purposes of this discussion, the birds-in-flight analogy will be used, although, in general, any collection of participants that exhibits this kind of group behavior falls under the label of "flocking." For example, flocking behavior is often used to control herds of animals moving over terrain. Of course, in this case, their motion is limited to the surface of a two-dimensional manifold. To use the flock analogy but acknowledge that it refers to a more general concept, Reynolds uses the term *boid* to refer to a member of the generalized flock. Much of this discussion is taken from his seminal paper [34].

There are two main urges at work in the members of a flock: *avoiding collision* and *staying part of the flock*. These are competing tendencies and must be balanced, with collision avoidance taking precedence.

Avoiding collisions is relative to other members of the flock as well as other obstacles in the environment. Avoiding collision with other members in the flock means that some spacing must be maintained between the members even though all members are usually in motion and that motion usually has some element of randomness associated with it to break up unnatural-looking regularity. However, because the objects, as members of a flock, are typically heading in the same direction, the relative motion between flock members is usually small. This facilitates maintaining spacing between flock members and, therefore, collision avoidance among members. The objective of collision avoidance should be that resulting adjustments bring about small and smooth movements.

Staying part of the flock has to do with each member trying to be just that—a member of the flock. In order to stay part of the flock, a member has an urge to be close to other members of the flock, thus working in direct opposition to the collision avoidance urge. In Reynolds' flock model, a boid stays part of the flock by a *flock centering force* [34]. But as he points out, a global flock centering force does not work well in practice because it prohibits flock splitting, such as that often observed when a flock passes around an obstacle. Flock centering should be a localized tendency so that members in the middle of a flock will stay that way and members on the border of a flock will have a tendency to veer toward their neighbors on one side. Localizing the flocking behavior also reduces the order of complexity of the controlling procedure.

Another force that is useful in controlling the member's reaction to movements of the flock is *velocity matching*, whereby a flock member has an urge to match its own velocity with that of its immediate neighbors. By keeping its motion relative to nearby members small, velocity matching helps a flock member avoid collision with other members while keeping it close to other members of the flock.

Local control

Controlling the behavior of a flock member with strictly local behavior rules is not only computationally desirable, it also seems to be intuitively the way that flocks operate in the real world. Thus, the objective is for control to be as local as possible, with no reference to global conditions of the flock or environment. There are three processes that might be modeled: *physics*, *perception*, and *reasoning and reaction*. The physics modeled is similar to that described in particle systems: gravity, collision detection, and collision response. Perception concerns the information about the environment to which the flock member has access. Reasoning and reaction are incorporated into the module that negotiates among the various demands produced as a result of the perception.

Perception

The main distinction between flocking and particle systems is the modeling of perception and the subsequent use of the resulting information to drive the reaction and reasoning processes. When one localizes the control of the members, a localized area of perception is modeled. Usually the "sight" of the member is restricted to just those members around it, or further to just those members generally in front of it. The position of the member can be made more precise to generate better defined arrangements of the flock members. For example, if a flock member always stays at a 45° angle, to the side and behind, to adjacent members, then a tendency to form a diamond pattern results. If a flock member always stays behind and to the side of one member with no members on the other side, then a **V** pattern can be formed. Speed can also affect perception; the field of view (fov) can extend forward and be slimmer in width as speed increases. To affect a localized fov, a boid should do the following:

Be aware of itself and two or three of its neighbors
Be aware of what is in front of it and have a limited fov
Have a distance-limited fov
Be influenced by objects within the line of sight
Be influenced by objects based on distance and size (angle subtended in the fov)
Be affected by things using an inverse distance-squared or distance-cubed weighting function
Have a general migratory urge but no global objective
Not follow a designated leader
Not have knowledge about a global flock center

Interacting with other members

A member interacts with other members of the flock to maintain separation without collision while trying to maintain membership in the flock. There is an attractive force toward other members of the flock while a stronger, but shorter range, repulsion from individual members of the flock exists. In analyzing the behavior of actual flocks, Potts [33] observed a *chorus line effect* in which a wave motion travels faster than any rate chained reaction time could produce. This may be due to perception extending beyond a simple closest-neighbor relationship.

Interacting with the environment

The main interaction between a flock member and the environment is collision avoidance, for which various approaches can be used. Force fields are the simplest to implement and give good results in simple cases. However, in more demanding situations, force fields can give undesirable results. The trade-offs of various strategies are discussed later, under the section Collision Avoidance.

Global control

There is usually a global goal that is used to control and direct the flock. This can be used to influence all members of the flock or to influence just the leader. The animation of the current leader of the flock is often scripted to follow a specific path or is given a specific global objective. Members can have a migratory urge, follow the leader, stay with the pack, or exhibit some combination of these urges.

Flock leader

To simulate the behavior of actual flocks, the animator can have the leader change periodically. Presumably, actual flocks change leaders because the wind resistance is strongest for the leader and rotating the job allows the birds to conserve energy. However, unless changing the flock leader adds something substantive to the resulting animation, it is easier to have one designated leader whose motion is scripted along a path to control the flock's general behavior.

Negotiating the motion

In producing the motion, three low-level controllers are commonly used. They are, in order of priority, collision avoidance, velocity matching, and flock centering. Each of these controllers produces a directive that indicates desired speed and direction (a velocity vector). The task is to negotiate a resultant velocity vector given the various desires.

As previously mentioned, control can be enforced with repulsion from other members and environmental obstacles and attraction to flock center. However, this has major problems as forces can cancel each other out. Reynolds refers to the programmatic entity that resolves competing urges as the *navigation module*. As Reynolds points out, averaging the requests is usually a bad idea in that requests can cancel each other out and result in nonintuitive motion. He suggests a prioritized acceleration allocation strategy in which there is a finite amount of control available, for example, one unit. A control value is generated by the low-level controllers in addition to the velocity vector. A fraction of control is allocated according to priority order of controllers. If the amount of control runs out, then one or more of the controllers receives less than what they requested. If less than the amount of total possible control is allocated, then the values are normalized (e.g., to sum to the value of one). A weighted average is then used to compute the final velocity vector. Governors may be used to dampen the resulting motion by clamping the maximum velocity or clamping the maximum acceleration.

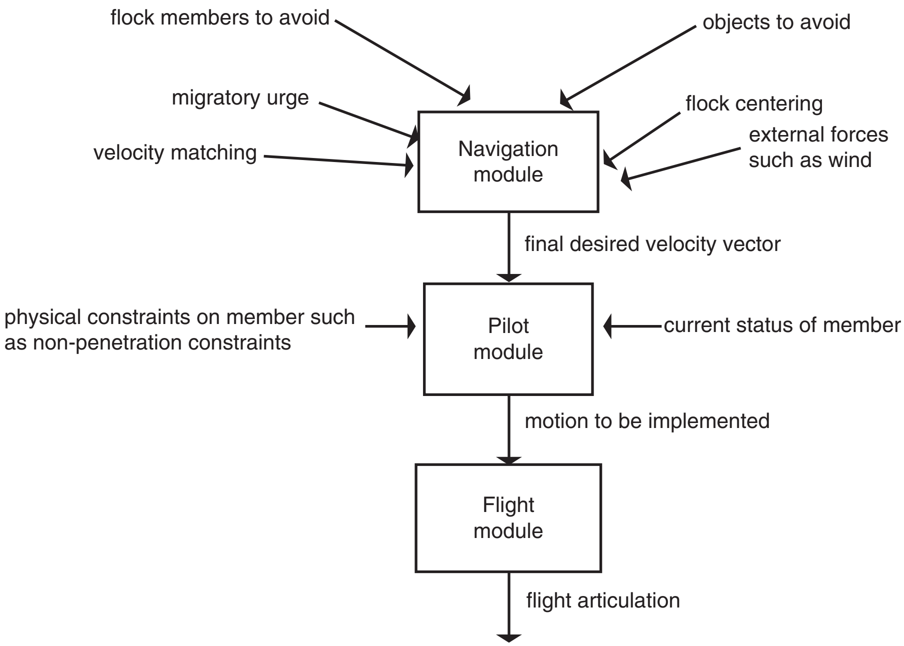


FIGURE 11.1

Negotiating the motion.

In addition to prioritized behaviors, the physical constraints of the flock member being modeled need to be incorporated into the control strategy. Reynolds suggests a three-stage process consisting of navigation, piloting, and flying (see Figure 11.1). Navigation, as discussed earlier, negotiates among competing desires and resolves them into a final desire. The pilot module incorporates this desire into something the flock member model is capable of doing at the time, and the flight module is responsible for the final specification of commands from the pilot module.

The navigation module arbitrates among the urges and produces a resultant directive to the member. This information is passed to the pilot model, which instructs the flock member to react in a certain way in order to satisfy the directive. The pilot model is responsible for incorporating the directive into the constraints imposed on the flock member. For example, the weight, current speed and acceleration, and internal state of the member can be taken into account at this stage. Common constraints include clamping acceleration, clamping velocity, and clamping velocity from below. The result from the pilot module is the specific action that is to be put into effect by the flock member model. The flight module is responsible for producing the parameters that will animate that action.

Collision avoidance

Several strategies can be used to avoid collisions. The ones mentioned here are from Reynolds's paper [34] and from his course notes [35]. These strategies, in one way or another, model the flock member's fov and visual processing. A trade-off must be made between the complexity of computation involved and how effective the technique is in producing realistic and intuitive motion.

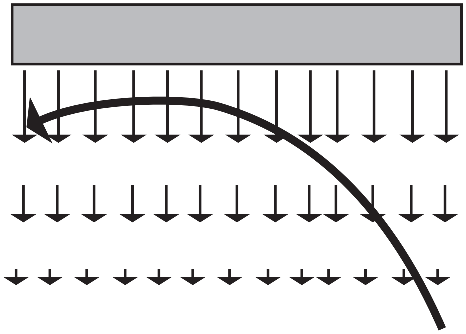


FIGURE 11.2

Force field collision avoidance.

The simple strategy is to position a limited-extent, repelling force field around every object. As long as a flock member maintains a safe distance from an object, there is no force imparted by the object to the flock member. Whether this occurs can easily be determined by calculating the distance[2] between the center point of the flock member and the center of the object. Once this distance gets below a certain threshold, the distance-based force starts to gently push the flock member away from the object. As the flock member gets closer, the force grows accordingly (see Figure 11.2). The advantages of this are that in many cases it effectively directs a flock member away from collisions, it is easy to implement, and its effect smoothly decreases the farther away the flock member is from the object, but it also has its drawbacks.

In some cases, the force field approach fails to produce the motion desired (or at least expected). It prevents a flock member from traveling close and parallel to the surface of an object. The repelling force is as strong for a flock member moving parallel to a surface as it is for a flock member moving directly toward the surface. A typical behavior for a flock member attempting to fly parallel to a surface is to veer away and then toward the surface. The collision avoidance force is initially strong enough to repel the member so that it drifts away from the surface. When the force weakens because of increasing distance, the member heads back toward the surface. This cycle of veering away and then toward the surface keeps repeating. Another problem with simple collision avoidance forces occurs when they result in a force vector that points directly toward the flock member. In this case, there is no direction indicated in which the member should veer; it has to stop and back up, which is an unnatural behavior. Aggregate forces can also prevent a flock member from finding and moving through an opening that may be more than big enough for passage but for which the forces generated by surrounding objects are enough to repel the member (see Figure 11.3).

The problem with a simple repelling force field approach is that there is no reasoned strategy for avoiding the potential collision. Various path planning heuristics that can be viewed as attempts to model simple cognitive processes in the flock member are useful. For example, a bounding sphere can be used to divert the flock member's path around objects by steering away from the center toward the rim of the sphere (Figure 11.4).

Once the flock member is inside the sphere of influence of the object, its direction vector can be tested to see if it indicates a potential intersection with the bounding sphere of the object. This calculation is the same as that used in ray tracing to see if a ray intersects a sphere (Figure 11.5).

[2]As in many cases in which the calculation of distances is required, distance-squared can be used thus avoiding the square root required in the calculation of distance.

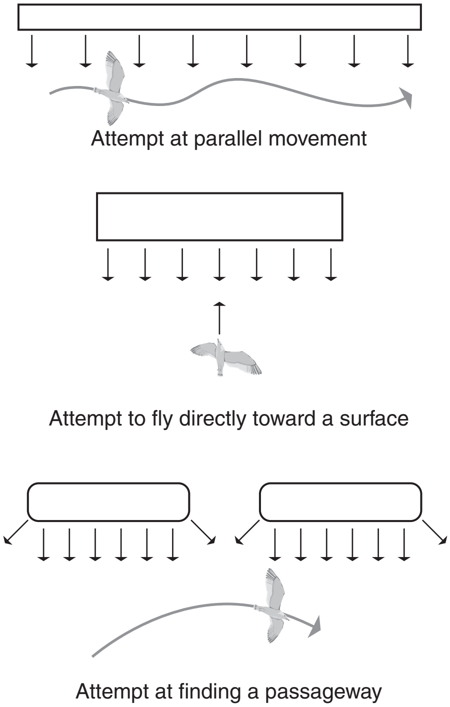


FIGURE 11.3

Problems with force field collision avoidance.

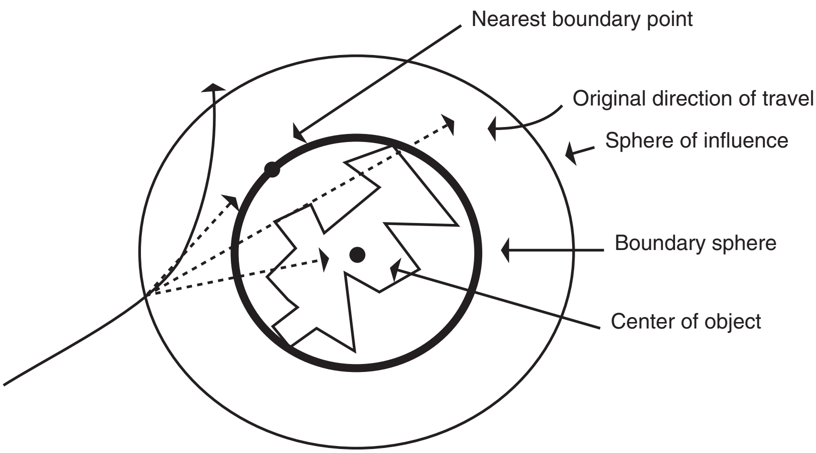
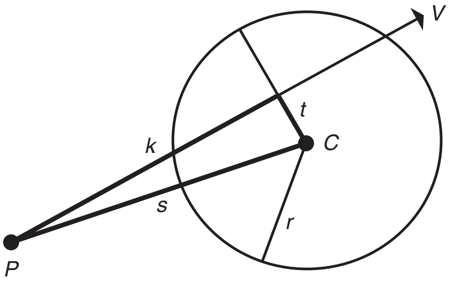


FIGURE 11.4

Steering to avoid a bounding sphere.



$$s = |C - P|$$

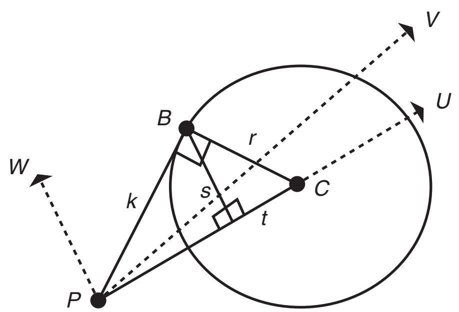
$$k = (C - P) \bullet \frac{V}{|V|}$$

$$t = \sqrt{s^2 - s^2}$$

$t < r$ indicates penetration with bounding sphere

FIGURE 11.5

Testing for potential collision with a bounding sphere.



$$k = \sqrt{|C - P|^2 - r^2}$$

$$r^2 = s^2 + t^2$$

$$k^2 = s^2 + (|C - P| - t)^2$$

$$k^2 = r^2 - t^2 + (|C - P| - t)^2$$

$$k^2 = r^2 - t^2 + (|C - P|^2 - 2|C - P|t + t^2$$

$$t = \frac{K^2 - r^2 - |C - P|^2}{-2|C - P|}$$

$$s = \sqrt{r^2 - t^2}$$

$$U = \frac{C - P}{|C - P|}$$

$$W = \frac{(U \times V) \times U}{|(U \times V) \times U|}$$

$$B = P + (|C - P| - t)\, U + sW$$

FIGURE 11.6

Calculation of point B on the boundary of a sphere.

When a potential collision has been detected, the steer-to-avoid procedure can be invoked. It is useful to calculate the point, B, on the boundary of the bounding sphere that is in the plane defined by the direction vector, V, the location of the flock member, P, and the center of the sphere, C. The point B is located so that the vector from P to B is in the plane defined earlier, is tangent to the sphere, and is on the same side of C as the vector from P to V (Figure 11.6).

Steering to the boundary point on the bounding sphere is a useful strategy if a close approach to the object's edge is not required. For a more accurate and more computationally expensive strategy, the flock member can steer to the silhouette edges of the object that share a back face and a front face with respect to the flock member's position and, collectively, define a nonplanar manifold. Faces can be tagged as front or back by taking the dot product of the normal with a vector from the flock member to the face. The closest distance between the semi-infinite direction of the travel vector to a silhouette edge can be calculated. If there is

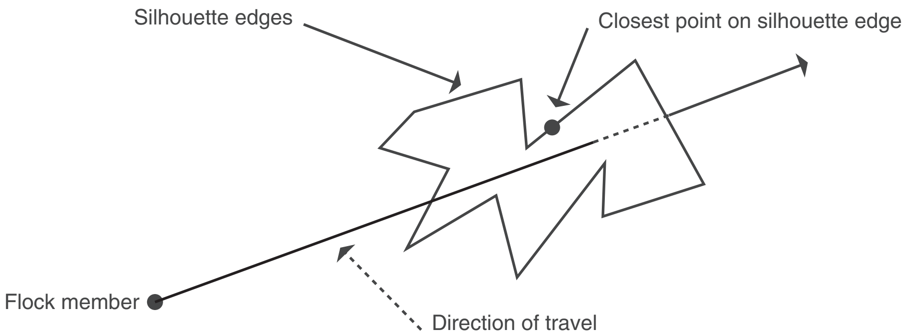


FIGURE 11.7

Determining steer-to-point from silhouette edges.

space for the flock member's body and a comfort zone in which to fly around this point, then this point can be steered toward that closest point on the silhouette (Figure 11.7).

An alternative strategy is to sample the environment by emitting virtual feelers to detect potential collision surfaces. The feelers can be emitted in a progressively divergent pattern, such as a spiral, until a clear path is found.

Another strategy is to model vision by projecting the environment to an image plane from the point of view of the flock member. By generating a binary image, the user can search for an uncovered group of pixels. Depth information can be included to allow searching for discontinuities and to estimate the size of the opening in three-space. The image can be smoothed until a gradient image is attained, and the gradient can be followed to the closest edge.

Splitting and rejoining

When a flock is navigating among obstacles in the environment, one of their more interesting behaviors is the splitting and rejoining that result as members veer in different directions and break into groups as they attempt to avoid collisions. If the groups stay relatively close, then flock membership urges can bring the groups back together to re-form the original single flock. Unfortunately, this behavior is difficult to produce because a balance must be created between collision avoidance and the flock membership urge, with collision avoidance taking precedence in critical situations. Without this precise balance, a flock faction will split and never return to the flock or flock members will not split off as a separate group; instead they will only individually avoid the obstacle and disrupt the flock formation.

Modeling flight

Since flocking is often used to model the behavior of flying objects, it is useful to review the principles of flight. A local coordinate system of *roll*, *pitch*, and *yaw* is commonly used to discuss the orientation of the flying object. The roll of an object is the amount that it rotates to one side from an initial orientation. The pitch is the amount that its nose rotates up or down, and the yaw is the amount it rotates about its up vector (see Figure 11.8).

Specific to modeling flight, but also of more general use, is a description of the forces involved in aerodynamics. *Geometric flight*, as defined by Reynolds [34], is the "dynamic, incremental, rigid transformation of an object moving along and tangent to a three-dimensional curve." Such flight is controlled by *thrust*, *drag*, *gravity*, and *lift* (Figure 11.9). Thrust is the force used to propel the object

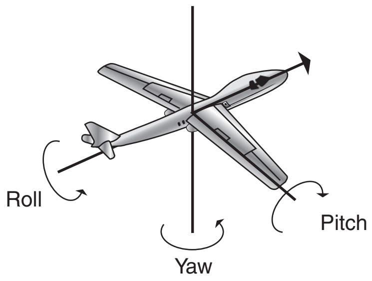


FIGURE 11.8

Roll, pitch, and yaw of local coordinate system.

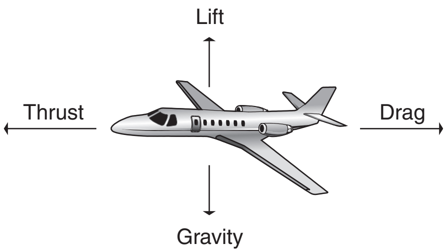


FIGURE 11.9

Forces of flight.

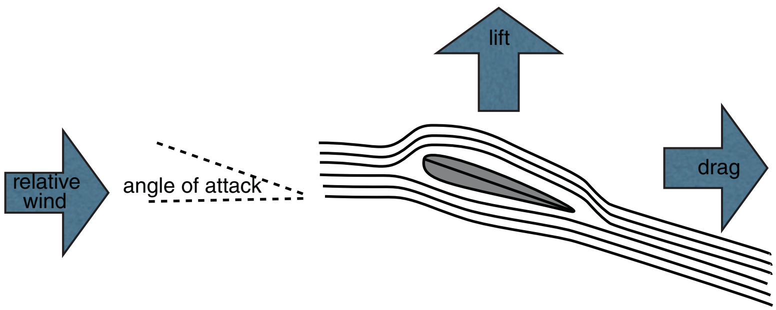


FIGURE 11.10

Lift produced by an airfoil.

forward and is produced by an airplane's engine or the flapping of a bird's wings. Drag is the force induced by an object traveling through a medium such as air (i.e., wind resistance) and works in the direction directly against that in which the object is traveling relative to the air. Gravity is the force that attracts all objects to the earth and is modeled by a constant downward acceleration. Lift is the upward force created by the wing diverting air downward. During flight, the airfoil's shape and orientation is such that air traveling over it is bent. A reaction to the change of momentum of the air is the lift (Figure 11.10). At higher angles of attack the downward velocity of the diverted air is greater. For straight and level flight, lift cancels gravity and thrust equals drag.

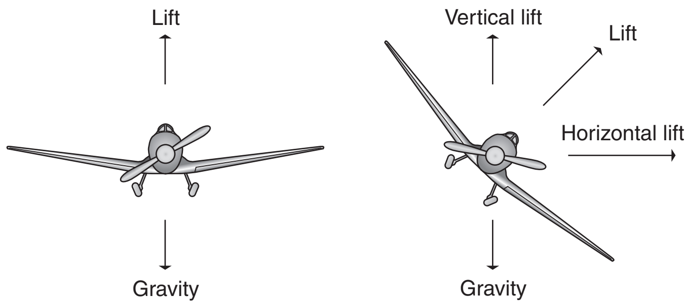


FIGURE 11.11

Lifting forces.

In flight, a turn is induced by a *roll*, in which the up vector of the object rotates to one side. Because the up vector is rotated to one side, the lift vector is not directly opposite to gravity. Such a lift vector can be decomposed into a vertical component, the component that is directly opposite to gravity, and the horizontal component, the component to the side of the object. A flying object turns by being lifted sideways by the horizontal component of the lift; this is why planes tilt into a turn. For the object to be directed into the turn, there must be some yaw induced. If a plane is flying level so that lift cancels gravity, tilting the lift vector to one side will result in a decrease in the vertical component of lift. Thus, to maintain level flight during a turn, one must maintain the vertical component of lift by increasing the thrust (see Figure 11.11).

Increasing the pitch increases the angle the wing makes with the direction of travel. This results in increased lift and drag. However, if thrust is not increased when pitch is increased, the increased drag will result in a decrease in velocity, which, in turn, results in decreased lift. So, to fly up, both thrust and pitch need to be increased.

These same principles are applicable to a soaring bird. The major difference between the flight of a bird and that of an airplane is in the generation of thrust. In a plane, thrust is produced by the propeller. In a bird, thrust is produced by the flapping of the wings.

Some of the important points to notice in modeling flight are as follows:

Turning is effected by horizontal lift
Turning reduces lift
Increasing pitch increases lift and drag
Increasing speed increases lift and drag

11.1.2 Prey–predator behavior

The previous discussion on flocking discussed modeling the very basics of intelligent behavior (not much beyond that of a college freshman): stay with your friends and avoid bumping into things. While flocking behavior can be interesting, especially when interacting with obstacles in the environment, the objective is to produce a single uniform motion—the emergent behavior of the flock. In order to create more interesting interaction among the members, two sets of characters can be created with competing goals and discriminating physical abilities.

The classic example of competing goals is that of a prey–predator model in which one set of characters is trying to eat the other. Very simple prey–predator models can create interesting animation if the rules of behavior are carefully chosen. Of course, if the two character groups have the same capabilities, then the resulting interaction may result in some pretty boring interaction. In general, interesting behavior might result when there are two or more classes of objects that have different qualities of motion so that, depending on the geometric arrangement of the characters and the environment, a member of one class or the other may win in a given situation. For example, one set of agents could be slower but have better acceleration, a better turning radius, a better turning velocity, and better reasoning capabilities. There is no absolute model to use when modeling character behavior because balancing the models is dependent on the desired motion.

At a minimum, the reasoning component for prey–predator models could be based on simple attraction/repulsion. One character class is attracted to the other (e.g., trying to eat it) while the other class is repulsed by the first (e.g., trying to avoid being eaten). While a simple force field model can produce interesting motion, incorporating some predictive reasoning ability in one or both character classes can make situational behavior more realistic. For example, the ability to compute the trajectory of a character from its current position and speed can produce more interesting and realistic behavior.

11.2 Knowledge of the environment

Behavioral animation is all about cognitive interaction with the environment combined with the limitations imposed by simulation of physical constraints. One of the important issues when modeling behavior is the information a character has access to about its environment.

At the simplest level, a character has direct access to the environment database. It has perfect and complete knowledge about its own attributes as well as the attributes of all the other objects in the environment. A character can "see" behind walls and know about changes to the environment well-removed from its locale. While this might be a handy computational shortcut, such knowledge can produce behaviors that are unrealistic. Characters reacting to events that are obviously hidden from view can destroy believability in the character.

More realistic behavior can be modeled if the character is enabled with locally acquiring knowledge about the environment. Such local acquisition is affected by simulating character-centric sensors, most commonly vision. Modeling vision involves determining what can be viewed from the position and orientation of the character. Modeling other senses might also be useful in some applications. Touch, for example, can be modeled by detecting collisions and might be useful for navigating through dark environments. Modeling sounds might be useful in a forest setting. However, since vision is by far the most common sense modeled in computer animation, the following discussion will be restricted to incorporating a character's sight into behavioral animation.

For a character that senses the environment, added realism can be created by providing the capability of remembering what has been sensed. Simulated memory is a way for senses to be recorded and accumulated. Current positions, motions, and display attributes (e.g., color) of other objects in the environment can be captured and used later when reasoning about the environment.

11.2.1 Vision

Vision systems limit knowledge about the environment by (1) modeling a limited fov and (2) computing visual occlusions. A simple form of vision only models the field of view and ignores occlusions. Using the character's position and (head) orientation, an fov can be easily calculated. Any object within that fov (e.g., by testing if the object's bounding box overlaps the fov), is marked as "visible" and, therefore, is known to the character.

While such fov is fast and easy to compute, it does not provide for appropriate behavior when significant objects or events are within the fov but occluded from view. Occlusions provide the possibility of hide-and-seek behaviors. Occlusion is provided by computing the geometric elements that are visible from the point of view of the characters and is modeled by rendering a scene from the character position. The vision system can deal with occlusions by incorporating (1) ray casting that samples the environment in order to determine the closest object along the ray and the associated distance or (2) a low-resolution z-buffer that pseudocolors objects with identification tags to indicate what object is visible at each of the pixels and the associated distance (z-depth).

For the most part in computer animation applications, the issues associated with image processing and perception are avoided by allowing the actor to access the object database directly once it has been determined that an object is visible (e.g., [13]). Thus, the actor immediately knows what the object is and what its properties are, such as its orientation and speed. The object's position in space can be retrieved from the ray casting depth or z-buffer. Alternatively, the object's position in space as well as other properties can be determined from the object database directly once it is determined that it is visible.

Vision does not help unless it tells the character something about its environment and lets the character modify its behavior accordingly. The simplest information available to the character is spatial occupancy—vision tells the character where things are. In a prey–predator model, enabling better vision can tip the scales in favor of a slower prey.

As an example of vision, Tu and Terzopoulos [41] have developed behavioral animation of artificial fishes with synthetic vision. A 300 degree fov with a radius consistent with translucent water is used. Any object within that field and not fully occluded is "seen." The artificial fish is given partial access to the object database in order to retrieve relevant information about the sighted object. In work by Noser et al. [24], as well as Kuffner and Latombe [18], a z-buffer vision system is used to determine visible objects for navigating through an obstacle-filled environment. The three-dimensional position of a given pixel is computed from the depth information in the z-buffer in Noser's work, whereas in Kuffner's work the object database is accessed to locate objects.

Obviously, reasoning plays a large part in any kind of intelligent behavior. Information about the environment is only as useful as what the agent can do with it. Prediction and evaluation are the primary building blocks of reasoning agents. Prediction allows the agent to guess what a current state will be. Evaluation allows the agent to estimate how good a particular situation is or to estimate how good a given situation will be.

11.2.2 Memory

Vision provides knowledge that is immediate but fleeting. If the only information available to the character is provided by the vision system, what is known is what is seen unless a mechanism exists to record what is seen for later use—memory. The simplest type of memory is accumulating a description of the static world. For example, a vision system that has depth information can record the world space coordinates of observed objects. A three-dimensional representation of the character's observed world can be built as the character traverses the environment. In work by Noser et al. [24], a quadtree data structure is used to partition space and construct an occupancy map from the vision information. As more of the environment is seen, more knowledge is accumulated.

Moveable objects create a problem for an accumulation-only occupancy map. If an object is only inserted and never removed from the occupancy map, then it will pollute the map if it moves throughout the scene and is observed in several places. Thus, dynamic objects demand that an update mechanism be included with the memory module. If an object is seen at a certain location, other references to its location must be removed from the occupancy map.

This raises the issue of the validity of the recorded location of a moving object that has been out of view for awhile. Chances are that if an object is seen moving at one location and then is out of view for a while, the object is no longer at that location. It seems like a good idea to remove memories of locations of moving objects if they have not been seen for awhile. Time-stamping memories of moving objects allows this to be done. As time progresses, the memories that are older than some (possibly object-specific) threshold are to be removed (e.g., [18] [24]).

11.3 Modeling intelligent behavior

Modeling behavior, especially intelligent behavior, is an open-ended task. Even simple flocking and prey–predator models can be difficult to control. Humans are even more complex and modeling their behavior is a task that AI has been addressing for decades. This type of AI falls under the rubric of computer graphics because of the spatial (geometric) reasoning component. In computer animation, the animator usually wants the visuals of character motion to mimic (or at least be a caricature of) how humans are seen behaving in real life. While motion capture is a relatively quick and easy way to model human behavior, developing algorithms to produce human behavior, if done well, are more flexible and more general than motion capture (mocap)-based animation. In addition, such models often contribute to a greater understanding of human behavior. Human and animal behavior have been modeled since the early days of computer animation but only recently have more sophisticated methods been employed for higher level, autonomous cognitive behavior.

For example, the Improv system of Ken Perlin and Athomas Goldberg uses layered scripts with decision rules to create novel animation [29]. The AlphaWolf project at Massachusetts Institute of Technology has modeled not only expressiveness, but also learning and development [40]. Attempts have also been made to apply neural net technology to learning behavior (e.g., [7]). Recently, there has been a more formal approach by referring to the literature in psychology (e.g., [36]). It has been applied to expressions and gestures, personality, and crowd behavior.

11.3.1 Autonomous behavior

Autonomous behavior, as used here, refers to the motion of an object that results from modeling its cognitive processes. Usually, such behavior is applied to relatively few objects in the environment. As a result, emergent behavior is not typically associated with autonomous behavior, as it is with particle systems and flocking behavior, simply because the numbers are not large enough to support the sensation of an aggregate body. To the extent that cognitive processes are modeled in flocking, autonomous behavior shares many of the same issues, most of which result from the necessity to balance competing urges.

Autonomous behavior models an object that knows about the local environment in which it exists, reasons about the state it is in, plans a reaction to its circumstances, and carries out actions that affect its environment. The environment, as it is sensed and perceived by the object, constitutes the external state. In addition, there is an internal state associated with the object made up of time-varying urges, desires, and emotions, as well as (usually static) rules of behavior.

It should not be hard to appreciate that modeling behavior can become arbitrarily complex. Autonomous behavior can be described as various levels of complexity, depending on how many and what type of cognitive processes are modeled. Is the objective simply to produce some interesting behavior, or is it to simulate how the actual object would operate in a given environment? How much and what kinds of control are to be exercised by the user over the autonomous agent? Possible aspects to include in the simulation are sensors, especially vision and touch; perception; memory; causal knowledge; commonsense reasoning; emotions; and predispositions. Many of these issues are more the domain of AI than of computer graphics. However, besides its obvious role in rendering, computer graphics is also a relevant domain for which to discuss this work because the objective of the cognitive simulation is motion control. In addition, spatial reasoning is usually a major component of cognitive modeling and therefore draws heavily on algorithms associated with computer graphics. Applications include military simulations (e.g., [21]), pedestrian traffic (e.g., [37]), and, of course, computer games (e.g., [15]).

Autonomous behavior is usually associated with animal-like articulated objects. However, it can be used with any type of object, especially if that object is typically controlled by a reasoning agent. Obvious examples are cars on a highway, planes in the air, or tanks on a battlefield. Autonomous behavior can also be imparted to inanimate objects whose behavior can be humanized, such as kites, falling leaves, or clouds. The current discussion focuses on the fundamentals of modeling human behavior.

Internal state

Internal state is modeled partially by intentions. Intentions take on varied importance depending on the urges they are meant to satisfy. The instinct to survive is perhaps the strongest urge and, as such, takes precedence over, say, the desire to scratch an itch. Internal state also includes such things as inhibitions, identification of areas of interest, and emotional state. These are the internal state variables that are inputs to the rest of the behavioral model. While the internal state variables may actually represent a continuum of importance, Blumberg and Galyean [1] group them into three precedence classes: imperatives, things that must be done; desires, things that should be done, if they can be accommodated by the reasoning system; and suggestions, ways to do something should the reasoning system decide to do that something.

Levels of behavior

There are various levels at which an object's motion can be modeled. In addition to that discussed by Blumberg and Galyean [1], Zeltzer [44] and Korein and Badler [17] discuss the importance of decomposing high-level goals into object-specific manipulation of the available degrees of freedom (DOFs) afforded by the geometry of the object. The levels of behavior are differentiated according to the level of abstraction at which the motion is conceptualized. They provide a convenient hierarchy in which the implementation of behaviors can be discussed. The number of levels used is somewhat arbitrary. Those presented here are used to emphasize the motion control aspects of autonomous behavior as opposed to the actual articulation of that motion (refer to Figure 11.12).

Internal state and *knowledge of the external world* are inputs to the *reasoning unit*, which produces a strategy intended to satisfy some objective. A *strategy* is meant to define the *what* that needs to be done. The *planner* turns the strategy into a *sequence of actions* (the *how*), which is passed to the *movement coordinator*. The movement coordinator selects the appropriate *motor activities* at the appropriate time. The motor activities control specific DOFs of the object.

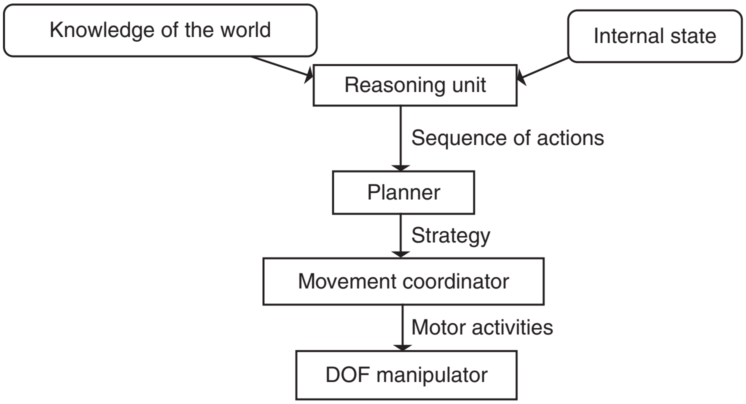


FIGURE 11.12

Levels of behavior.

Keeping behavior under control

One of the fundamental concerns with autonomous behavior, as with all high-level motion control, is how to model the behaviors so that they are generated automatically but are still under the control of the animator. One of the motivations for modeling behaviors is to relieve the animator from specifying the time-varying values for each of the DOFs of the underlying geometry associated with the object. Pure autonomy is often not the ultimate goal in designing autonomous agents; often behavior needs to be controlled in order to be relevant to the objectives of the animator. Also, the control needs to occur at various levels of specificity.

The various levels of behavior provide hooks with which control over behavior can be exercised. The animator can insert external control over the behavior by generating imperatives at any of the various levels: strategies, action sequences, and/or activity invocation. In addition, more general control can be exercised by setting internal state variables.

Arbitration between competing intentions

As with arbitration among competing forces in flocking, behaviors cannot, in general, be averaged and expected to produce reasonable results. The highest precedent behavior must be accommodated at all costs, and other behaviors may be accommodated if they do not contradict the effect of the dominant behavior. One approach is to group behaviors into sets in which one behavior is selected. Selected behaviors may then be merged to form the final behavior.

11.3.2 Expressions and gestures

Nothing is more boring than a speaker who stands up and dryly repeats a script without expression, without a personality. As we have seen in Chapter 10, there has been a fair amount of work on so-called talking heads. These are animations where a synthetic face lip-syncs a script. Most of these are cartoon-like in their lip movement, but they use facial expressions to add interest to the animation.

Several "talking heads" have been created that include facial expressions coordinated with synthetic speech. *Greta* [31] includes wrinkles and furrows to enhance realism. *Ruth*, the Rutgers

University Talking Head [5] [6], animates nonverbal signals in synchrony with speech and lip movements. The intonation is specified using the tones and break indices (ToBI) standard [27]. Brow expressions are categorized in terms of Ekman's facial action units [11]. The expressions are generated by rules inferred from lexical structure along with observed behavior of live performances. Additionally, in face-to-face communication the eyes alone can convey information about the participants, beliefs, goals, and emotions [30].

Upper body gestures (i.e., arm and torso) allow more opportunity for expressive speech. *BEAT*, a toolkit by Justine Cassell et al. [2], allows animators to type in text they wish spoken by a synthetic figure. Using linguistic and contextual information contained in the text, the movements of the hands, arms, and face and intonation of the speech can be controlled. A knowledge base relating gestures to emotions within the context of grammatical constructions is used to suggest gestural behaviors. These are passed through an activity filter that produces a schedule of behaviors that is converted to word timing and associated gestural animation. Laban movement analysis (LMA) (e.g., [19]) is used as a theoretical framework for generating the qualitative aspects of linguistic gestures by Chi et al. in the *EMOTE* system [4]. The shape and effort components of LMA are used to control the timing of the gesture, the articulation of the arms during the gesture, and animation of the torso. As an alternative to parameterized models of behavior, data-driven techniques use captured human motion to animate a conversational character [3] [38]. Performance data are segmented and annotated in order to provide a database of motion and speech samples. These samples are recombined and blended together to create extended utterances.

11.4.3 Modeling individuality: personality and emotions

In the previous section, the ability to generate gestures along with speech makes a character more believable. In the EMOTE system, the gestures can be modified by shape and effort parameters [4]. While in EMOTE these values are externally supplied, it demonstrates the ability to vary behavior based on parameters. By making these parameters associated with the internal state of a character, individuals within a population can be differentiated by the quality of their behavior by their *personality* [43]. For the current discussion, personality refers to the time-invariant traits that control the idiosyncratic actions of an individual. *Emotions* are considered time-varying attributes of an individual. *Mood* is sometimes considered a third attribute of individuality, and is a personal trait that is time varying but on a longer timescale than emotion.

Graphics research into modeling personalities often borrows from research in psychology, where there are already models proposed for characterizing personalities. These models include OCEAN and PEN. The OCEAN model of personality, also called the Five-Factor Model, or the "Big 5," consists of openness, conscientiousness, extroversion, agreeableness, and neuroticism [32]. The PEN model of Hans Eysenck has three dimensions of personality: extraversion, neuroticism, and psychoticism [14]. The Eysenck Personality Profiler is widely used in business. It is a questionnaire that measures 21 personality traits along the three dimensions.

As with personality, graphics research in emotions has borrowed heavily from psychology. Psychology, however, is not definitive when defining basic emotions and what those basic emotions are. As identified by Ekman [10], there are six basic emotions: happy, sad, fear, disgust, surprise, and anger. Emotional expressions and gestures contribute to communication (vis-á-vis linguists). Five basic emotions are suggested by Oatley and Johnson-Laird [25]: happiness, anxiety, sadness, anger, and disgust. Ekman [12] has also suggested there are 15 emotions: amusement, anger, contempt, contentment, disgust, embarrassment, excitement, fear, guilt, pride in achievement, relief, sadness/distress, satisfaction, sensory pleasure, and shame. The Ortony Clore Collins (OCC) model describes how the perceptions of people dictate how they experience emotions [28].

The Improv system [29] provides tools for an animator to script the behavior of characters that respond to users and to other characters in real time. It consists of a behavior engine that is responsible for executing, in parallel, scripts that are applicable to the current situation. The scripts allow for layered behavior so that the animator can build high-level behaviors on top of low-level ones. It provides for nondeterministic behavior by including probabilistic weights to be associated with alternative choices. Underneath the behavior engine is the animation engine, which is responsible for procedurally generating motion segments and blending between them. Personality and emotions are not explicitly modeled but must be encoded in the choices and probabilities supplied with the scripted behavior.

The *AlphaWolf* project of Tomlinson and Blumberg [40] models three parameters of emotion: pleasure, arousal, and dominance. These are updated based on past emotion, an emotion's rate of drift, and environmental factors. In addition, there is context-specific emotional memory that can bring a character back to a previously experienced emotion.

Egges et al. described a system based on the OCC model [8] [9]. The big five personality traits are used to modify emotional states. An intelligent three-dimensional talking head with lip-sync speech and emotional states creates behavior. Emotions are used to control the conversational behavior of an agent. A method is employed to update a character's emotional state based on personality, emotions, mood, and emotional influences.

11.4 Crowds

A crowd is a large group of people physically grouped (crowded) together. But the term "crowd" has been applied to a wide variety of scenarios, such as a collection of individuals reacting to the environment (e.g., [42]), a mass of hundreds of thousands of armed warriors in a feature length film where individual identification is practically impossible, and dots on a screen showing movement from room to room during a fire simulation. For this discussion, a crowd is considered to be the animation of representations of a large number of individuals where the individuals are not the primary focus of the animation and detailed inspection of individuals by the viewer is not expected. This distinguishes a crowd from an environment containing multiple agents in which the individual activity of the agents is important. The use of multiple agents is just that—some number of behaviorally animated characters where the actions of individuals are discernible and important and the viewer would be inspecting the activity of individuals in the environment. The emphasis here is on animation of multiple figures where the global nature of the activity is the primary consideration.

There are two main applications of graphical crowd modeling. The first is as a visual effect. Perhaps the obvious one is the use in feature length film to fill in the movement or activity of a mass of people in battle scenes or populating an environment with activity or creating an audience in the bleachers of a stadium. In battle scenes, for example, it is common to have the first row or two of characters to be live actors while all the background activity is synthetic characters. Managing the complexity of such crowds has become an important issue in the visual effects industry and is being addressed by software products such as Massive [20] [23]. Computer games are another obvious example of this application.

In this type of crowd, the appearance and motion of the individuals must be organized to be believable, not distracting, and computationally efficient.

The second application of graphical crowd modeling is as a simulation of activity in a space in order to better understand that activity. Here, the visuals are not the focus. The distribution and spatial relationship of crowd members with respect to each other and with respect to the environment are the main concern. For example, crowd animation might be used to model the translocation of a mass of people, such as a crowd entering an amusement park or exiting a stadium, in order to evaluate the various pathways through an environment for traffic flow. A related use that has received a fair amount of attention recently is evacuation simulation. In these cases, the appearance of the individuals has no value, it is only the location of the members and how they traverse the environment while trying to escape some threat, such as fire, usually in a building. Here, the cognitive modeling and subsequent data gathering are the important tasks. The visuals are either not important at all or are of secondary importance. Often the crowd members are represented merely as dots in a bird's-eye view of a building floor plan.

11.4.1 Crowd behaviors

There are certain behaviors that individuals of a crowd assume because they are members of the crowd. These behaviors are similar to that found in members of a flock: collision avoidance and maintaining crowd membership. Simulating crowd behavior is both an interesting and challenging research area. Using Reynolds' flock of boids as a reference, crowd members engage in collision avoidance, match the velocity of neighbors, and stay close to other crowd members. For example, in finding seats in a theater or evacuating a building, the flow of the crowd can be studied [16]. However, as opposed to typical models of flocking behavior, crowd flow can be multidirectional. Using psychology literature as a source for how people behave in crowds (e.g., [36]), one of the main activities found in crowds is the avoidance of oncoming people when traversing an environment. Various rules can be derived by looking at actual behavioral studies, as seen in the following:

- In high density, start avoiding at 5 feet or nearer; in low density, avoidance can start at 100 feet
- In low density, change paths; in high density, rotate body and side step (*step and slide*)
- Open (front to other person, men) versus closed (back to other person, women)
- In changing path, move to open side else tend to move to right
- Avoid people moving in the same direction by slowing down or overtaking them and resuming the original course after the pass has been completed

11.4.2 Internal structure

There are higher cognitive behaviors that create substructures within the group. While a crowd can exhibit an aggregate behavior (e.g., so-called *mob mentality*) that is similar to the flocking behavior of birds and emerges from localized motion of the flock members, it is usually made up of smaller groups of people in which the group members interact closely. Group membership is dynamic, and group interaction often occurs [22].

Groups are formed by common urges (e.g., going for lunch), belief systems (e.g., political allies), and emotional state (e.g., soccer fans) as well as by spatial proximity. Individuals of a crowd may or may not have a group membership. Qualities of an individual relevant to group membership include the following:

- A changeable belief system: From experiences, senses (e.g., vision) understanding the nature of things as a result of learning, and causes and effects (from seeing or experiencing a sequence of events)
- Time-varying biases, propensities, disposition, inclination, tendency, and emotional state: Things that will be satisfied if possible and if there are enough extra DOFs in attaining a goal or desire to satisfy secondary desires
- Goals, desires: Definite objectives that the agent is trying to satisfy

11.4.3 Crowd control

Similar to flocking behavior, individual crowd members are typically represented as being aware of nearby crowd members and must be managed to avoid collisions. As opposed to flocking behavior, there is less concern about the effect of physics on the crowd members, because crowds are often land-based animals (e.g., humans) where collisions are avoided rather than detected and movement occurs in two-dimensional space. However, because of the large number of members typically found in a crowd, strategies for controlling the motion of each member must be very efficient for many applications. Three main strategies have been used to control crowds: rules, forces, and flows. Rules have been discussed under the topic of intelligent behavior. For crowds, usually simple rules are used in order to keep the computation cost low. Forces can be used that are similar to those found in particle systems but must be organized so as to prevent collisions. The work by Helbing (e.g., [45]) concerning what he calls *social forces* is basically an extension of Reynolds' work on flocking behavior that is applied to social interactions in crowds. In the third approach to crowd control, the movement of crowd members has been compared to flow fields. This is usually effective for crowds that are massive in the number of members and only positional information of an individual is needed.

11.4.4 Managing *n*-squared complexity

Because of the typically large number of members in a crowd, the task of comparing each crowd member to every other crowd member for discovering the spatial proximity relationship is a major concern. As a consequence, some type of spatial decomposition approach has often been used in cases where the domain of interest is known ahead of time, for example, in building evacuation simulations. In such cases, a spatial sort can be used to sort the crowd members into a data structure conducive for identifying neighbors within a certain distance. For example, a grid-based approach can be used in which all of the interesting space is broken down into a rectangular matrix of cells. With a small enough cell size, each cell holds one group member so that movement into a cell is restricted to empty cells. This can be useful in maintaining the "personal space" of a crowd member, for example. Of course this also results in a large number of cells, so updating the occupancy of each cell can start to incur an undesirable overhead. Increasing the cell size reduces the number of cells, but requires maintaining a list of crowd members inside of each cell and increases the complexity of finding the nearest neighbor in a given direction. Other structures can also be used such as binary space partitioning, a hierarchical quadtree, or dynamic Voronoi diagrams, but these require modification as the crowd members change location over time.

11.4.5 Appearance

In cases where the appearance of individuals is of interest (e.g., feature films), the objective is to produce a population with enough variability so the viewer is not distracted. If every individual in the crowd had exactly the same appearance, it would noticeable to, and therefore distracting for, the viewer. The same holds for detectable patterns in appearance. The objective here is to not have individual features, or relationships among features, of the crowd noticeable to the casual viewer. A common strategy is to create mutually exclusive, yet compatible, sets with variations. For appearance, the sets might be hair color, hair style, shirt color, shirt type, pants color, and pants type. By combinatorial arithmetic, it is easy to see that even just 2 variations per set results in 64 different individual appearances. This same approach can be taken for the movements of the individuals. For example, walking behavior can be divided into the following sets: gait length, walking speed, arm swing distance, frequency of head swivel, and frequency of head nods. Pseudorandomness can be an important part of breaking noticeable patterns in the crowd. Even with the same gait, introducing slight phase shifts in the walk can improve the visual impression of the crowd.

Because crowds can be viewed at a variety of distances, they lend themselves to geometric level-of-detail modeling. The term *impostor* has been used to mean an alternate representation of a graphical element that is of reduced complexity and is used in place of the original geometry when viewed from a distance. Impostors can be used to reduce the display complexity of crowd members at large viewing distances. Crowd members also lend themselves to level-of-detail behavioral modeling [26]. At large viewing distances, the motion of the member can be simplified as well as the appearance. For example, walking can be represented by a series of sprites (two-dimensional graphical objects overlaid on the background) that have simple representations of leg and arm swing behavior. Because individual behaviors are arbitrarily complex, there are opportunities to abstract out "typical crowd behavior" in order to simplify modeling of the individuals (if the viewing conditions permit) when the behavior of the individual member is only statistically relevant (e.g., [39]). Actions can be selected on a statistical basis and strung together to create the visual effect of a crowd.

11.6 Chapter summary

We are only beginning to understand the complexities of spatial reasoning, idiosyncratic behavior, and generic behavior. There is no firm theoretical footing on which to stand when modeling personalities, emotions, and memory like when modeling physical interactions of rigid objects. This represents both a challenge and an opportunity for researchers and developers in computer animation.

References

[1] Blumberg J, Galyean T. Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments. In: Cook R, editor. Computer Graphics. Proceedings of SIGGRAPH 95, Annual Conference Series. Los Angeles, Calif: Addison-Wesley; August 1995. p. 47–54. ISBN 0-201-84776-0.

[2] Cassell J, Vilhjalmsson H, Bickmore T. BEAT: the Behavior Expression Animation Toolkit. In: SIGGRAPH.

[3] Cassell J, Pelachaud C, Badler N, Steedman M, Achorn B, Becket T, et al. ANIMATED CONVERSATION: Rule-Based Generation of Facial Expression, Gesture & Spoken Intonation for Multiple Conversational Agents. In: SIGGRAPH 94.

[4] Chi D, Costa M, Zhao L, Badler N. The EMOTE Model for Effort and Shape. In: Computer Graphics. Proceedings of SIGGRAPH 2000. New Orleans, La.; 23–28 July; 2000. p. 173–82.

[5] DeCarlo D, Stone M. The Rutgers University Talking Head: RUTH. In: Rutgers Manual. www.cs.rutgers.edu/~**village/ruth/ruthmanual10.pdf.**

[6] DeCarlo D, Revilla C, Stone M. Making Discourse Visible: Coding and Animating Conversational Facial Displays. In: Proceedings of Computer Animation; 2002. p. 11–6.

[7] Dinerstein J, Egbert P, de Garis H, Dinerstein N. Fast and Learnable Behavioral and Cognitive Modeling for Virtual Character Animation: Research Articles. Computer Animation and Virtual Worlds May 2004;15(2).

[8] Egges A, Kshirsagar S, Magnenat-Thalmann N. A Model for Personality and Emotion Simulation. In: Knowledge-Based Intelligent Information & Engineering Systems (KES2003). 2003. p. 453–61.

[9] Egges A, Kshirsagar S, Magnenat-Thalmann N. Generic Personality and Emotion Simulation for Conversational Agents. Computer Animation and Virtual Worlds January 2004;15(1):1–13.

[10] Ekman P. Universals and Cultural Differences in Facial Expressions of Emotion. In: Cole J, editor. Nebraska Symposium on Motivation 1971, vol. 19. Lincoln, NE: University of Nebraska Press; 1972. p. 207–83.

[11] Ekman P. About Brows: Emotional and Conversational Signals. In: von Cranach V, Foppa K, Lepenies W, Ploog D, editors. Human Ethology: Claims and Limits of a New Discipline: Contributions to the Colloquium. Cambridge: Cambridge University Press; 1979. p. 169–202.

[12] Ekman P. Basic Emotions. In: Dalgleish T, Power M, editors. Handbook of Cognition and Emotion. Sussex, U.K.: John Wiley & Sons, Ltd.; 1999.

[13] Enrique S, Watt A, Maddock S, Policarpo F. Using Synthetic Vision for Autonomous Non-Player Characters in Computer Games. In: 4th Argentine Symposium on Artificial Intelligence Santa Fe, Argentina; 2002.

[14] Eysenck H. Biological Dimensions of Personality. In: Pervin L, editor. Handbook of Personality: Theory and Research. New York: Guilford; 1990. p. 244–76.

[15] Isla D, Blumberg B. New Challenges for Character-Based AI in Games. In: Artificial Intelligence and Interactive Entertainment: Papers from the 2002 AAAI Spring Symposium. AAAI Press; 2002.

[16] Johnson C. The Glasgow-Hospital Evacuation Simulator: Using Computer Simulations to Support a Risk-Based Approach to Hospital Evacuation, http://www.dcs.gla.ac.uk/~johnson/papers/G-HES.PDF**; Sept. 2005.**

[17] Korein J, Badler N. Techniques for Generating the Goal-Directed Motion of Articulated Structures. IEEE Computer Graph Appl November 1982.

[18] Kuffner J, Latombe JC. Fast Synthetic Vision, Memory, and Learning Models for Virtual Humans. In: Proceedings of Computer Animation; 1999. p. 118–27.

[19] Maletic V. Body, Space, Expression: The Development of Rudolf Laban's Movement and Dance Concepts. New York: Mouton de Gruyte; 1987.

[20] Massive Software. Massive Software, http://www.massivesoftware.com/massive.php**; Santa Fe, Argentina 2007.**

[21] McKenzie F, Petty M, Kruszewski R, Gaskins R, Nguyen Q-A, Seevinck J, et al. Crowd Modeling for Military Simulations Using Game Technology. In: SCS International Journal of Intelligent Games & Simulation (IJIGS). 2004.

[22] Musse S, Thalmann D. A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis. In: Computer Animations and Simulations '97. Proc. Eurographics Workshop. Budapest: Springer Verlag, Wien; 1997. p. 39–51.

[23] New Zealand Trade & Enterprise. Crowd Control, http://www.nzte.govt.nz/common/files/la2-massive.pdf**; February 2007.**

[24] Noser H, Renault O, Thalmann D, Magnenat-Thalmann N. Navigation for Digital Actors Based on Synthetic Vision, Memory and Learning. Computer and Graphics 1995;19(1):7–19.

[25] Oatley K, Johnson-Laird P. Towards a Cognitive Theory of the Emotions. Cognition and Emotion 1:29–50.

[26] O'Sullivan C, Cassell J, Vilhjalmsson H, Dingliana J, Dobbyn S, McNamee B, et al. Levels of Detail for Crowds and Groups. Computer Graphics Forum 21(4):733–42.
[27] The Ohio State University Department of Linguistics. ToBI, http://ling.ohio-state.edu/~tobi/, **Sept. 2005.**
[28] Ortony A, Clore G, Collins A. The Cognitive Structure of Emotions. Cambridge, U.K.: Cambridge University Press; 1988.
[29] Perlin K, Goldberg A. Improv: A System for Scripting Interactive Actors in Virtual Worlds. In: Rushmeier H, editor. Proc. SIGGRAPH 96. New York: ACM Press; 1996. p. 205–16.
[30] Poggi I, Pelachaud C. Signals and Meanings of Gaze in Animated Agents. In: Prevost S, Cassell J, Sullivan J, Churchill E, editors. Embodied Conversational Characters. Cambridge, MA: MIT Press; 2000.
[31] Pelachaud C, Poggi I. Subtleties of Facial Expressions in Embodied Agents. Journal of Visualization and Computer Animation 2002;13(5):301–12.
[32] Popkins N. The Five-Factor Model: Emergence of a Taxonomic Model for Personality Psychology, www.personalityresearch.org/papers/popkins.html**; Sept. 2005.**
[33] Potts W. The Chorus Line Hypothesis of Maneuver Coordination in Avian Flocks. Nature May 1984;309:344–5.
[34] Reynolds C. Flocks, Herds, and Schools: A Distributed Behavioral Model. In: Stone MC, editor. Computer Graphics. Proceedings of SIGGRAPH 87, vol. 21(4). Anaheim, Calif.; July 1987. p. 25–34.
[35] Reynolds C. Not Bumping into Things. In: Physically Based Modeling Course Notes. SIGGRAPH 88. Atlanta, Ga.; August 1988. p. G1–G13.
[36] Rymill S, Dodgson N. A Psychologically-Based Simulation of Human Behavior. In: Theory and Practice of Computer Graphics. 2005. p. 25–42.
[37] Shao W, Terzopoulos D. Autonomous Pedestrians. In: Symposium on Computer Animations, Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Los Angeles; 2005. p. 19–28.
[38] Stone M, DeCarlo D, Oh I, Rodriguez C, Stere A, Lees A, et al. Speaking with Hands: Creating Animated Conversational Characters from Recordings of Human Performance. In: Proceedings of SIGGRAPH. 2004.
[39] Sung M, Gleicher M, Chenney S. Scalable Behaviors for Crowd Simulation. Computer Graphics Forum 2004;23(3), Eurographics '04.
[40] Tomlinson B, Blumberg B. Social Behavior, Emotion and Learning in a Pack of Virtual Wolves. In: AAAI Fall Symposium "Emotional and Intelligent II: the Tangled Knot of Social Cognition". North Falmouth, MA; November 2–3, 2001.
[41] Tu X, Terzopoulos D. Artificial Fishes: Physics, Locomotion, Perception, Behavior. In: Computer Graphics. Proc. of SIGGRAPH 94. p. 43–50.
[42] Ulicny B, Thalmann D. Crowd Simulation for Interactive Virtual Environments and VR training Systems. In: Proceedings of the Eurographic workshop on Computer Animation and Simulation Table of Contents. Manchester, UK; 2001. p. 163–70.
[43] Wong K. Personality Model of a Believable and Socially Intelligent Character. M.Sc Thesis. University of Sheffield; 2004.
[44] Zeltzer D. Task Level Graphical Simulation: Abstraction, Representation and Control. In: Badler N, Barsky B, Zeltzer D, editors. Making Them Move. San Francisco: Morgan Kaufmann; 1991.
[45] Helbing D, Molnár P. Social Force Model for Pedestrian Dynamics. Physical Review E 1995;51(5):4282–6.