Fish Schooling and Shoaling Simulation

Jim Pickrell [1] **Adrien Hadj-Chaib** [1] **Tomoyori Iwao** [1] **Wenqi Zou** [1]

Abstract

In this simulation, we modeled the schooling behavior of horse eyed jacks, a common fish in the Caribbean off Mexico. We first simulated behavior for a single fish, then added more fish to create a school. We also disrupted the school with coral obstacles, shrimp (food) and sharks (danger). We were happy with the results. The simulated movements and responses of the fish are appropriate and realistic.

1. Introduction

Schooling and shoaling are the behaviors of groups of fish. "Schooling" implies they are swimming in an organized formation. "Shoaling" means a group of fish that are associating together but are not exhibiting group traveling behavior. In other words, they are staying together for social reasons but are not going anywhere.

Schooling is an intriguing behavior because schools often act like a single organism, exhibiting group intelligence and group decision making, yet the group has no leader. Schooling behaviors can be found in many kinds of fish, as well as birds, herds of mammals, bees and other insects, and many other animals. Simulations of schooling are useful as it enables people to better understand fish behaviors. It can also be applied to games and movies where simulations of large numbers of people or animals are required.

Rules employed in our simulation of fish schooling behaviors are borrowed from Reynolds' theories(Reynolds, 1987). To mimic reality, we also added shrimp, which serves as fish food, sharks, which are perceived as danger by fish, and corals as shown in Fig. 1.

The simulation is written in JavaScript using the UCLA Tiny Graphics library. It is essentially a game that the viewer can play online in a web browser. Random terrain is automatically generated, and the viewer can add fish, shrimp, or sharks. Added creatures will interact with each other. Their behaviors are evaluated based on proximity to reality.

[1] Computer Science Department, University of California, Los Angeles, Los Angeles, CA, USA.

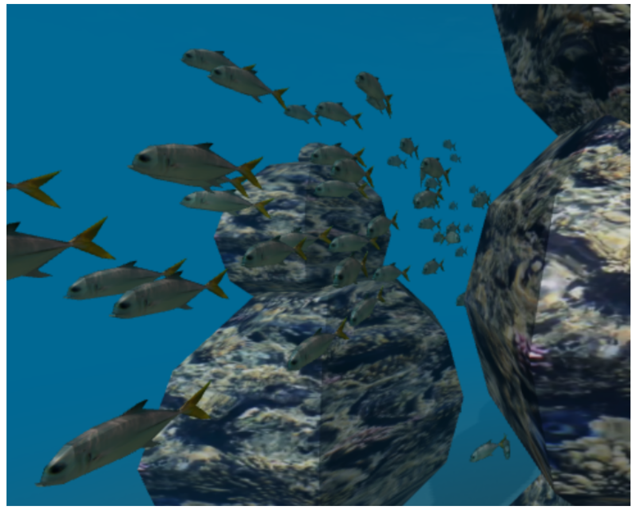

Figure 1. Simulated school of fish traveling

The simulation may be seen online: `http://www.brandx.net/cs275/`

Actual videos of fish on the Palancar reef location in Mexico were taken and used to compare against simulated fish behaviors. We found that the simulated behavior to be a good match for the real fish. The simulation looks the way we expected it to look and is similar to the video.

2. Related Work

Our simulation is based on two articles by Craig Reynolds proposing simple rules to explain the theoretical basis for autonomous movement.

In the first article, Reynolds describes three behaviors that make schooling behavior possible. Reynolds names the behaviors and assigns each one an acceleration. (Reynolds, 1987)

In the second article, Reynolds expands theories on the general "steering" of individual "vehicles". For example, a "flee" behavior will accelerate a fish away from a predator. He provides us with a simple logical framework to analyze these behaviors. Schooling is simply a special application of the more general "steering" rules. (Reynolds, 1999)

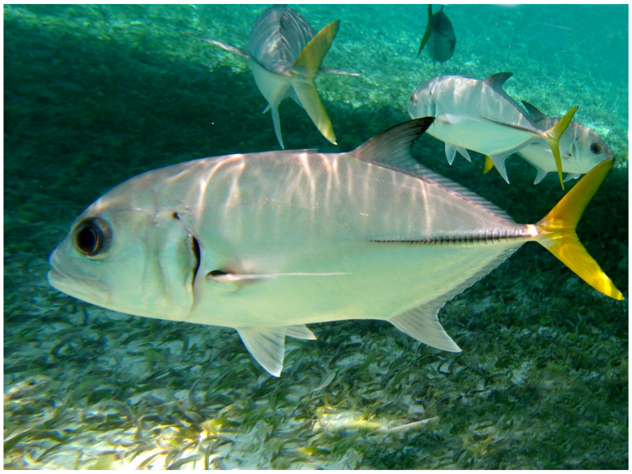

Figure 2. Horse eye jack

3. Fish, Sharks and Shrimp

To create a model of schooling behavior, we created models for fish, sharks, and shrimp. Each has slightly different behavior. We modelled them after the real fish found in the Palancar Reef off Cozumel Island in Mexico.

3.1. Horse Eye Jacks

Horse eye jacks (Fig. 2) are about two feet long, and eat shrimp and small fish. The name comes from their very large eyes. They travel in schools of ten to several hundred members. Jacks are one of the most common fish in the Mexican Carribean, and other warm water areas. They are chosen because they are typical of other schooling fish, common, and easy to observe.

3.2. Nurse Sharks

Nurse sharks are the most common type of shark on the Palancar Reef. They are typically 7-10 feet long, and eat smaller fish as well as lobsters and other food found at the sea bottom. They have long fins, a very flexible tail, and a broad head that almost looks like a catfish. Nurse sharks are not fast moving and are considered to be lazy sharks. Our shark was modelled using photographs of real nurse sharks.

3.3. Shrimp (Copepod)

Horse Eye Jacks eat shrimp as well as various small fish. One of the most common plankton shrimp is the Copepod, which generally float and eat smaller plant plankton. Copepods are hugely important in the ocean food chain. They have one of the largest biomasses on the planet. Copepods don't really swim much, but they rise during the night to feed and sink during the day to avoid being seen by predators.

We colored our shrimp bright magenta and made them larger than the real ones, so they would be easy to see.

4. Fish Steering Theory

We simulated a school of fish by modeling each fish as an autonomous unit that follows Reynolds' theories. Reynolds suggests that units in a simulation be treated as "vehicles," which is acted upon by the following forces:

- Thrust: forward acceleration
- Braking: deceleration or reverse thrust
- Steering: pushes us left or right

The implementation of above behaviors is achieved by combining the following forces:

4.1. Separation

Real fish keep a minimum distance from each other to avoid collisions. In our project, we applied a separation force on fish to avoid crowding other fish. We also have a variable for the preferred distance between two fish. Changing this variable changes the spacing of the school.

To find the separation force $\overrightarrow{\theta}_{separation}$ on fish f_k:

$$\overrightarrow{\theta}_{separation} = A * \frac{\sum_{i=1}^{m} \overrightarrow{V_i}}{m} \tag{1}$$

where $\overrightarrow{V_i}$ is a normalized vector from fish F_i to F_k
m is the number of fish within distance d from the fish k, and $m \geq 1$
A is a proportional constant

4.2. Alignment

Fish in a school want to align with each other. We steer our fish towards the average heading of local fish by applying a alignment force on them.

The weight factor for an alignment force controls how important an alignment-induced direction change is to the fish. Some fish (sharks for example) have a fairly low priority on alignment, and some have none at all (shrimp). This alignment force is calculated by averaging their orientation directions of fish/shark nearby.

To find the alignment force $\overrightarrow{\theta}_{alignment}$ on fish f_k:

$$\overrightarrow{\theta}_{alignment} = A * \frac{\sum_{i=1}^{m} \overrightarrow{V_i}}{m} \tag{2}$$

where $\overrightarrow{V_i}$ is a normalized vector velocity for fish F_i
m is the number of fish within distance d from the fish k,

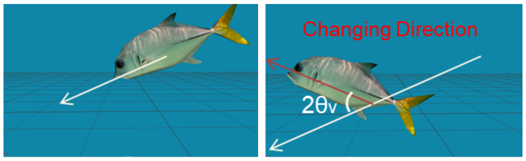

Figure 3. Fish Leveling

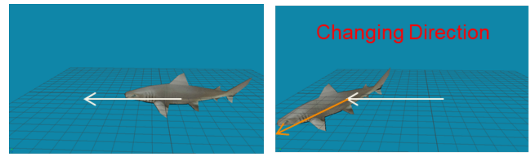

Figure 4. Shark Wandering

and $m \geq 1$
A is a proportional constant

4.3. Cohesion

Fish in a school like to swim with other fish. We coded a cohesion force on each fish to steer it to move towards the average position of local fish. This is counterbalanced by the "separation" rule which will overrule if the fish get too close.

To find the cohesion force $\overrightarrow{\theta}_{cohesion}$ on fish f_k:

$$\overrightarrow{\theta}_{cohesion} = A * ((\frac{\sum_{i=1}^{m} \overrightarrow{P_i}}{m}) - \overrightarrow{P_k}) \tag{3}$$

where $\overrightarrow{P_i}$ is the position for fish f_i
m is the number of fish within distance d from the fish k, and $m \geq 1$
A is a proportional constant

4.4. Collision Avoidance

Fish, sharks and shrimp need to avoid obstacles such as coral and the sea floor. This is done by adding a small acceleration perpendicular to the path and away from the obstacle.

To do the calculation, we will assume that both fish $\overrightarrow{P1}$ and obstacle $\overrightarrow{P2}$ are spheres of radius r1 and r2 with distance d from center to center. So, the distance d1 between fish and obstacle is

d1 = d - r1 - r2

Next, we find the point of nearest approach $\overrightarrow{P3}$ where our travel path path is closest to the obstacle $\overrightarrow{P2}$.

The direction from target center $\overrightarrow{P2}$ to point of nearest approach $\overrightarrow{P3}$, is our deflection acceleration $\overrightarrow{A}$.

$\overrightarrow{A}$=$\overrightarrow{P3}$-$\overrightarrow{P2}$

It's a very simple calculation, but gives a good result. The deflection vector should be scaled for velocity, changing time, and distance from the obstacle. We will add this deflection vector to our velocity, and our path will go around the obstacle.

4.5. Avoiding Danger

Real life fish will flee from any sharks they see. This instinctual survival is simulated in our project. If there are any any sharks nearby, a steering force is applied to the fish so that it swims away from sharks.

To find the fleeing force $\overrightarrow{\theta}_{danger}$ on fish f_k:

$$\overrightarrow{\theta}_{danger} = A * (\frac{\sum_{i=1}^{m} \overrightarrow{P_k} - \overrightarrow{P_i}}{m}) \tag{4}$$

where $\overrightarrow{P_i}$ is the position for shark f_i
m is the number of sharks
A is a proportional constant

If a fish sees one shark, it will head directly away from it. If there is more than one shark, it will try to run away from all of them.

A school of fish that has spotted a shark will move according to the interactions stated in previous sections. The entire school will swim in a manner that avoids nearby sharks, which shows group decision making.

4.6. Leveling

Fish can swim in any direction, but prefer to swim horizontally. As shown in Fig. 3, they change their vertical direction slightly depending on time t. If they swim at a certain vertical degree for several seconds t_s, they change their vertical direction. We call this leveling. Given θ_o as an angle, which is decided by other forces such as separation and alignment, θ_v as an vertical angle where fish changes periodically for leveling, and θ_y, which is an actual fish swimming angle, then we have

$$\theta_y = \begin{cases} \theta_o + \theta_v, & if \quad (2n+1)t_s \leq t < 2(n+1)t_s \\ \theta_o - \theta_v, & if \quad 2nt_s \leq t < (2n+1)t_s \end{cases}$$

where $n \in \mathbb{Z} \cap n \geq 0$.

4.7. Wandering

Sharks seek food. Sharks change their swimming direction θ if they cannot find food. We set the time threshold t_{th} when sharks decide to change their swimming direction and implement shark wandering as shown in Fig. 4.

4.8. Combining Forces

Each behavior generates an acceleration. The forces are combined, using weighting factors, and then added sequentially.

The velocity will be scaled to the maximum velocity after each addition, which means the last force added will override any previous forces.

In our simulation the behaviors are evaluated in this order:
- levelling
- wandering
- schooling
- chasing food
- avoiding danger
- avoiding obstacles

This means levelling is the least important behavior, while avoiding danger and avoiding obstacles are the most important.

5. Simulated Behaviours

In addition to fish schooling, as explained in the previous sections, we also simulated other sea life behaviors.

5.1. Shrimp Jumping

Copepods are very sensitive to small variations in pressure. When a fish approaches a shrimp, the shrimp jumps a few inches to avoid being eaten. The fish closest to the shrimp won't catch it. After the shrimp jumps, it needs to recharge before jumping again. This means other fish will have a good chance of catching it.

A shrimp within bite distance of a fish is considered eaten and removed from play.

5.2. Shark Eating

If a fish is within the vision distance of a shark, the shark opens its mouth and a force is applied on the shark in the direction of that fish.

If a fish is within bite distance of the shark, the fish is considered eaten and removed from play.

6. Modeling and Animation

In this chapter, we will introduce CG modeling by Maya. Maya is a popular 3d animation program from AutoDesk. Subsequently, we will describe the structure of the opening mouth animation based on two object files created in Maya.

6.1. Geometric Modeling

In this section, we will illustrate how to create a geometric model with Maya using the nurse shark as an example. As shown in Fig. 5, to model a shark, we project a real image to a 2D plane in Maya's 3D space. After that, by using an orthographic camera to the 2D plane, we make a cube shape with subdivisions and the modify it to match the shark. We keep the models simple and minimize the number of polygons to make rendering easy.

6.2. UV mapping

After geometric modeling, we conduct a UV-mapping to the shapes for rendering a texture by tiny graphics. There are several methods to make a good UV on a surface.

The first method we use is a plane UV-mapping as shown in Fig. 6. This image is directly referenced from Maya's support page `https://knowledge.autodesk.com/support/maya`. In this method, we project a planar UV to the shapes. After that we fix the UV coordinates because planar UV is not perfectly fit to the shape's UV. By using the UV-editor in Maya as shown in 7, we can fix the UV easily. By using this graphical User Interface, we can change a UV coordinate of each vertex not to make a mismatch between the texture and the shape. Plane UV-mapping is suitable if the shape is symmetry and there is a picture that is taken at the side of the object. By using the picture, we only conduct a UV-mapping to a half part of the shape. Therefore, we use a plane UV-mapping by using the picture of fish and a shark taken at the side of them.

For the spherical coral obstacles, we used a sphere UV-mapping as shown in Fig. 8. This image is also directly referenced from Maya's support page. As shown in this Figure, sphere UV-mapping is suitable for spherical objects.

6.3. Animation

Using the shark model, we created its mouth animation. When a shark eating its foods, the shark opens its mouth. We define this shark's mouth state $S_m = 1\{open\}$. On the other hand, when it seeks foods, it closes their mouth. We define this shark's mouth state $S_m = 0\{close\}$. We implement this animation by setting the distance threshold d_f. When the distance between a shark and the nearest fish is longer than the threshold, the shark will close its mouth. When the distance is shorter, the shark opens the mouth to eat foods. Thus, defining the position of the shark p_s and the position of the k_{th} foods p_{f_k}, we have

$$S_m = \begin{cases} 1, & if \quad \underset{f_k}{\operatorname{argmin}} ||p_s - p_{f_k}|| \le d_f \\ 0, & if \quad \underset{f_k}{\operatorname{argmin}} ||p_s - p_{f_k}|| > d_f \end{cases}$$

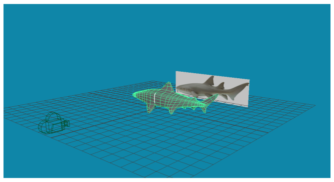

Figure 5. Shark Modeling

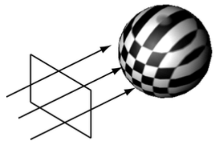

Figure 6. Plane UV-mapping

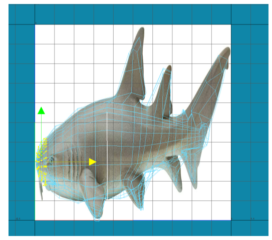

Figure 7. UV editing

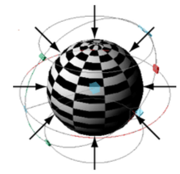

Figure 8. Sphere UV-mapping

7. Tiny Graphics Improvement

The Tiny Graphics JavaScript library is an Object Oriented interface for creating 3D graphics in WebGL. We worked with the developers of this library to use several new features that were useful for this project.

7.1. Instancing

Many of the objects in our scenes, such as fish and coral, are repeated. A school can be rendered at once using instancing, greatly reducing rendering overhead. Instancing is a graphics card feature that became available using the newest version of Tiny Graphics. Instancing lets us improve performance by minimizing the number of low-level drawcalls, translating into a fuller and more immersive scene.

7.2. Texture Mapping

The new version of Tiny Graphics has better .obj model loading as well as support for .mtl material specification generated by Maya. These new texture mapping capabilities also included accurate texture coordinate loading, and better Phong shading yielding brighter colors. These improvements allowed us to easily import textured Maya models to Tiny Graphic, and render fish with photographic textures easily.

7.3. Fog

The blue fog effect was implemented as a fragment shader specifically for this project. It is not a part of Tiny Graphics. It was achieved by changing the color buffer clearing color to blue, and interpolating between the fragment color and the clearing color using the distance of the fragment the camera. The fog effect allows us to simulate underwater loss of visibility and hide the borders of the floor and ceiling planes.

7.4. Shadows

The new Tiny Graphics version has native support for real-time shadow mapping for both directional and point-lights. This scene uses a single directional light casting a shadow over the entire scene. This allows the creation of interesting effects such as having shadows follow the movements of the fish. The light source was also given an oscillatory movement to simulate the effects of surface waves on light direction. This makes the shadows of all objects oscillate, adding to the scene's realism, providing a better three dimensional feeling.

8. Evaluation

We evaluated our simulation by comparing it to real life sea animal videos.

8.1. Planacar Reef Videos

Jim Pickrell, Jens Gronemeyer, and Gary Moyer recorded a number of videos of fish schooling on the Planacar Reef off Cozumel Island in April 2021. The equipment used was Olympus underwater camera in a Sea Frogs underwater housing, which allowed depths up to about 200 feet. Some samples are listed below.

Fish schooling 4 Palancar
`https://www.youtube.com/watch?v=nzSSPJdnnbU`

Fish Schooling, Cozumel
`https://www.youtube.com/watch?v=g6by6fIZ8kE`

Fish Schooling 447 Palancar Reef
`https://youtu.be/0oMNpcluyAI`

Fish Schooling 441 Palancar Reef
`https://youtu.be/jZCXQr8XTdc`

Fish schooling 440, Palancar Reef
`https://youtu.be/1LvQVdE1Ttg`

8.2. Simulation Details

The viewing area of our simulation presents an endless aquarium. Viewers are allowed to add fish, sharks, and shrimp, and watch as they interact. There are several buttons at the bottom: Add Food (adds a pink shrimp at a random location), Add Shark, Add Fish, Add a School of fish (adds ten fish at a random location), Clear Fish (removes all fish

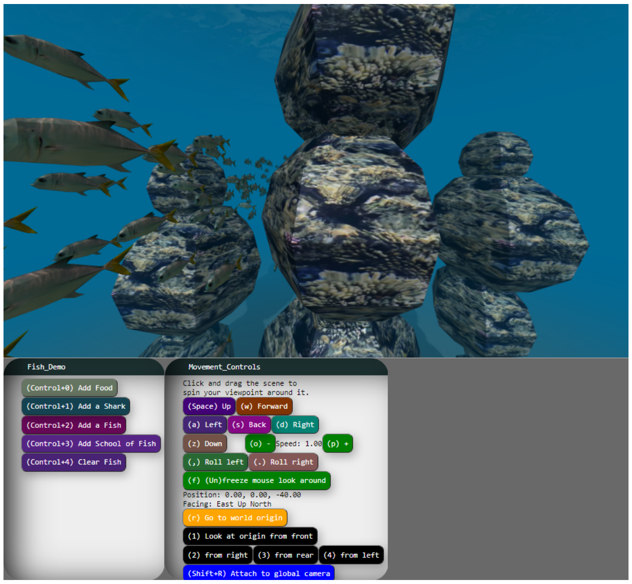

Figure 9. Simulation UI

from the scene). There are also several buttons that allow viewers to alter the point of view of the camera. Fig. 9 shows the UI of our simulation.

8.3. Evaluation Result

The following behaviors are simulated in our project:
- Fish swim alone
- Fish swim as school
- Fish avoid obstacles
- Fish eat food
- Shark chases fish
- Large numbers of fish onscreen at the same time

The simulated fish match the behavior shown in the videos. Alone, our simulated fish swim in a similar fashion as the real fish. In a school, our fish swim close to other fish but never collide, as shown in Fig. 10. When danger appears, the fish scatter, but will reform a school again as soon as they can.

Obstacle avoidance is a success. Both fish and sharks avoid collisions with coral. When their path is blocked, a school of fish will go to one side or the other of a coral, or may even go to both sides and reform on the other side.

Our simulated fish eating behavior looks close to reality. A fish that sees a shrimp will eat it, or at least try. Sometimes the shrimp are able to get away by jumping, but if several fish attack at the same time, the shrimp will probably be eaten.

Our sharks are really designed to chase the fish so they can

Figure 10. The above picture is captured from a real fish video. The below picture is captured from our simulation

exhibit schooling behavior. We set the speed of fish and sharks to be the same to provide balance to the simulation and make sure all the fish were not eaten immediately.

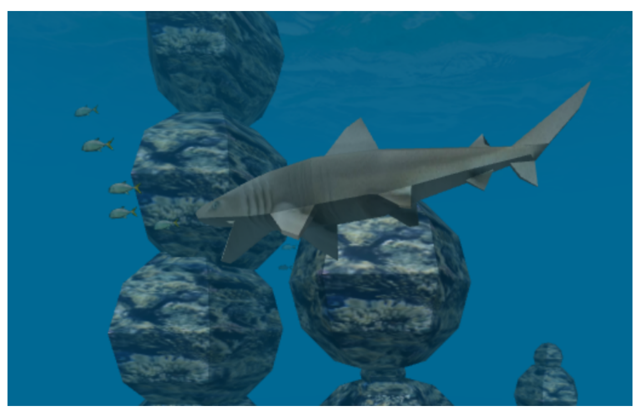

Figure 11. Shark chases fish

Our simulation allows hundreds of fish to school at the same time. We can simulate large-scale schooling behavior as the fish encounter food, sharks and obstacles. Simulations with 500 to 1000 fish remain realistic and run smoothly, as shown in Fig. 12.

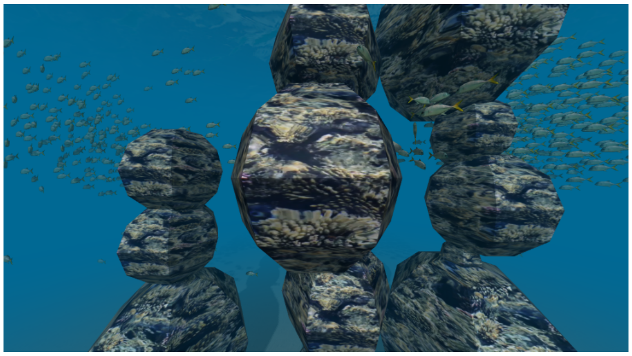

Figure 12. Simulation with 500 fish

In conclusion, the simulated fish schooling behaviors closely match what we see in the videos.

We compared the way that the fish travel from one place to another, the way they scatter when disturbed, and the way they travel around obstacles. These simulations seem accurate when compared to our videos.

9. Conclusion

We achieved our goals with this simulation: We have built a live, interactive model of schools of fish that looks realistic and can be easily modified to model other types of fish with other behaviors. Reynolds' theories provided us with a good model for schooling behaviors. We have compared our simulation to videos of real fish, and our simulation is close to real fish behaviors captured in the video.

In the future, we would like to spend more time studying the behaviors of fish on the Palancar reef so that we can further improve the accuracy of our simulation. We would also like to improve the quality of the graphics and animation, improve the simulation of coral terrains and sea surfaces, and add additional types of sea life.

In particular, the behavior of sharks is worth further investigation and simulation.

It would also be interesting to see if we can use this model to explain why fish travel in schools. Some speculate that there is safety in numbers. Others suggest that it makes it easier to find food. There are also many other theories. Further development of this simulation would allow us to explore those questions.

References

Reynolds, C. W. Flocks, herds and schools: A distributed behavioral model. New York, NY, USA, 1987. Association for Computing Machinery. ISBN 0897912276.

Reynolds, C. W. Steering behaviors for autonomous characters. In *Proceedings of Game Developers Conference 1999 held in San Jose, California. Miller Freeman Game Group, San Francisco, California*, pp. 763–782, 1999. URL `http://www.red3d.com/cwr/papers/1999/gdc99steer.html`.