Unified Motion Planner for Fishes with Various Swimming Styles

Daiki Satoi*[1] Mikihiro Hagiwara†[1] Akira Uemoto‡[1] Hisanao Nakadai§[1] Junichi Hoshino¶[1]

[1]Entertainment Computing Lab, University of Tsukuba

Abstract

We propose a unified motion planner that reproduces variations in swimming styles based on the differences in the fish skeletal structures or the variations in the swimming styles based on changes in environmental conditions. The key idea in our method, based on biology, is the following. We considered the common decision-making mechanism in fish that allows them to instantly decide "where and how to swim." The unified motion planner comprises two stages. In the first stage, where to swim to is decided. Using a probability distribution generated by integrating the perceptual information, the short-term target position and target speed are decided. In the second stage, how to swim is decided. A style of swimming that matches the information for transitioning from the current speed to the target speed is selected. Using the proposed method, we demonstrate 12 types of CG models with completely different sizes and skeletal structures, such as manta ray, tuna, and boxfish, as well as a scene where a school of a few thousand fish swim realistically. Our method is easy to integrate into existing graphics pipelines. In addition, in our method, the movement characteristics can easily be changed by adjusting the parameters. The method also has a feature where the expression of an entire school of fish, such as tornado or circling, can be designated top-down.

Keywords: fish swimming, motion planning, fish school control

Concepts: •**Computing methodologies** → **Procedural animation;**

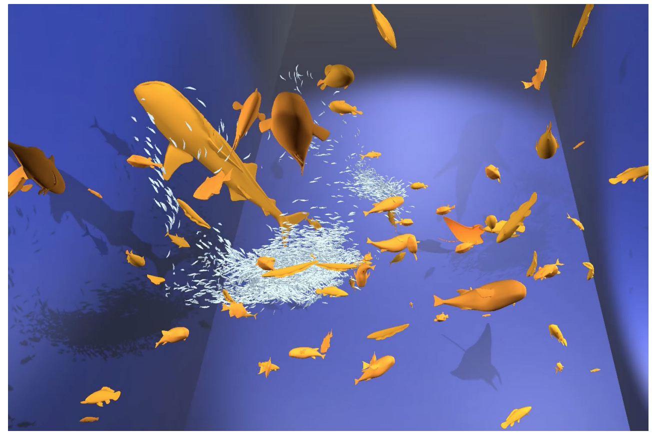

Figure 1: *A scene of a giant fish tank with 8,000 fish and 12 species of fish. The fish movements are simulated using the proposed method.*

1 Introduction

An expression of underwater scenes where many fish swim in a lively manner under water is necessary in animation [Stanton and Unkrich 2003], games [Yamaguchi 2008], and many other types of content.

Regarding fish, there is a very high level of diversity, e.g., 28,000 species [Nelson 2006]. Even when restricting the scope to a few ocean areas, as many as 83 species for Wakasa Bay, Japan [Masuda 2008] and 150 species for Hanalei Bay, Hawaii [Friedlander and Parrish 1997] have been confirmed. Fish inhabit diverse environments from shallow to deep seas and from tropical to polar oceans, and their body structure and swimming style widely vary [Lindsey 1978]. Furthermore, in some cases, fish will change their swimming style depending on the conditions; for example, they will use a swimming style called C-start when escaping from a predator [Domenici and Blake 1997].

Therefore, to realistically depict underwater scenes, we have to accurately recreate the variations in the swimming styles by considering the changes in the conditions in the fish environment. Recreating this kind of variation in swimming styles using key frame animation while considering large numbers of fish in an underwater scene is difficult. Furthermore, the animator must be able to easily show the characteristics of the different swimming styles, such as slow or swift swimming with quivering movements, the limitless freedom of fish swimming, and the fish swimming in circles in the same location.

To solve this problem, we concentrated on the mechanism of motion planning that actual fish perform in order to swim. In recent years, significant progress has been made in deciphering the decision-making ability of fish while swimming. For example, the archerfish and machaca will first watch the movements of the target when eating insects or falling fruit. Then, they will instantly make decisions on when and where to move and decide how much power to exert when starting to swim [Krupczynski and Schuster 2008; Schuster 2012]. In addition, in the field of fish physiology, it is known that fish, such as the wrasse or boxfish, drastically change the way they use their bodies and fins. In other words, they change their swimming style based on their swimming speed [Archer and Johnston 1989; Walker 2000; Hove et al. 2001].

Based on observations of fish biology or fish physiology, the instant decision making comprises two stages. First, there is the decision of destination, speed, etc. Then, there is the decision of the swimming style. This behavior is repeated while swimming and is common to many fish. Therefore, we propose a unified motion planner that models this common mechanism. In the first stage, where to swim to is decided. Using a probability distribution generated through the integration of perceptual information, the short-term target position

*e-mail:dsatoi@acm.org
†e-mail:hagiwara.mikihiro@entcomp.esys.tsukuba.ac.jp
‡e-mail:uemoto.akira@entcomp.esys.tsukuba.ac.jp
§e-mail:nakadai.hisanao@entcomp.esys.tsukuba.ac.jp
¶e-mail:jhoshino@esys.tsukuba.ac.jp

SIGGRAPH '16 Technical Paper, July 24-28, 2016, Anaheim, CA
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925977

and target speed are decided. In the second stage, how to swim is decided. A style of swimming that matches the transitioning from the current speed range to the target speed range is selected.

Using this method, the variations in swimming styles based on fish species or environmental changes can easily be made by changing the control rules for each swimming style. In Figure 1, a scene is shown where 12 species of fish, such as whale shark, tuna, manta ray, and boxfish, with completely different sizes and skeletal structures are simultaneously swimming. Moreover, the proposed method has the following characteristics:

1. The computational cost is low; therefore, as in Figure 1, large-scale schools of fish with a few thousands or more fish can be depicted as in a real ocean setting.

2. Using Tube-Following, which is a new route-setting method, the shape and spread of an entire school of fish, e.g., tornado or circling, can be shown.

3. The proposed method can be directly integrated into existing graphics pipelines. As long as generic fish CG models can be prepared with rigging already in place, the action of the CG models is realistically dictated by simply setting the parameters.

2 Related Work

2.1 Simulated Swimmers

Many methods have been proposed for the natural animation of the characters' swimming motions. For example, a method of generating realistic human swimming animation using a Central Pattern Generator (CPG), which generates basic repeated action patterns, was used in walking or swimming and biomechanical models [Si et al. 2014]. Another example involves the Evolving Virtual Creatures [Sims 1994], which succeeded in creating virtual creatures that move and behave in simulated 3D physical worlds using genetic algorithms. To create animations of schools of fish, bird-like objects (Boids), an artificial life program [Reynolds 1987], is often used. In the simplest Boids world, the emergent group behavior is accomplished using three simple rules.

Regarding methods dealing with fish-type characters, Artificial Fishes [Tu and Terzopoulos 1994; Terzopoulos et al. 1994; Grzeszczuk and Terzopoulos 1995] and Articulated Swimming Creatures [Tan et al. 2011] have been proposed. In Artificial Fishes, the fish-types are autonomous agents with sensors, motors, and learning and control functions. By modeling the physical properties of fish and the underwater environment as well as behavior such as grouping, escaping, and feeding, fish movements close to that of real fish are successfully recreated. Furthermore, Terzopoulos et al. have also proposed a general purpose method where users can easily model high-level cognitive behavior [Funge et al. 1999]. In Articulated Swimming Creatures, aquatic organisms of various shapes, such as fish and turtles, were successfully made to swim together using two-way solid-fluid coupling and optimizing the motion parameters.

In these pioneering studies, realistic swimming animation is accomplished by modeling the physical interaction between characters and the underwater environment. Our contribution in relation to these methods is to introduce a new motion planning method regarding fish swimming action and thus to reproduce the wide variety of swimming styles observed in real fish. In addition, with regard to fish and the single action of swimming, we think that high-level behavior, such as group behavior or escape behavior, and physiological properties, such as muscles, are important. Therefore, in the proposed method, the various effects on swimming are treated as manipulations of a probability distribution that chooses short-term movement targets. Consequently, swimming animation that satisfies multiple dynamically shifting constraints, such as "escape as quickly as possible while maintaining the group and without unrealistic acceleration," becomes possible. Virtual fish move randomly within a range that satisfies the constraints, and this allows for a natural variety of movements in a scene where many virtual fish are swimming together.

2.2 Character Control Framework

For human- and animal-type characters, numerous character control models that aim to reproduce natural movements have been proposed [Geijtenbeek and Pronost 2012]. These models can be roughly divided into example-based models, simulation-based models, or a combination of the two models.

Example-based methods are represented by cases where a wave-like gesture input from the user is robustly estimated and retargeted for character animation [Rhodin et al. 2015] or where sparse motion capture data are used to control the action of the wing beat of a bird [Ju et al. 2013].

Simulation-based methods comprise methods where three types of muscle fiber models and deformable characters with the choice of shapes move like living beings [Tan et al. 2012], or where the use of reinforcement learning and movement control on nonflat terrain are accomplished [Peng et al. 2015].

Both example- and simulation-based methods wherein the motion parameters are optimized or where realistic deformation is adopted using biomechanical models have become mainstream.

We focused on the singular task of fish swimming, where a wide variety of skeletal movements exist. The motion planning procedure of first deciding the destination and then the swimming style allowed for the realistic animation of various fish-type characters, even when using a very simple method of skeletal movement control.

2.3 Motion Planner

In the CG field, numerous motion-planning methods have been proposed for the posture control of characters [Wang et al. 2015] and global path planning [Kallmann and Kapadia 2014]. In addition, in robotics, motion planners have long been studied with motion planning methods that are based on neural networks, genetic algorithms, and Particle Swarm Optimization (PSO) [Masehian and Sedighizadeh 2007].

The motion planners aim for character posture control or global path planning; thus, it is difficult to use them for quick, agile changes in movement or changes in swimming style. Our method focuses on the local motion planning of the short-term and repeated movements of fish while swimming.

3 Swimming Mechanism in Real Fishes

In this chapter, prior to explaining the unified motion planner itself, we first explain a number of concepts that are important to our method.

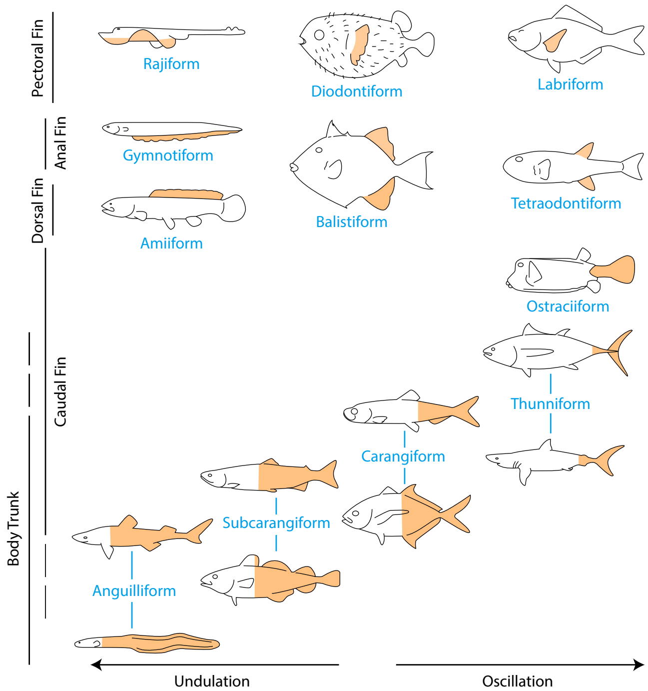

Figure 2: *12 types of swimming modes. The orange areas denote the primary movement regions. The vertical axis represents the types of primary movements and the horizontal axis represents the breadth of the primary movement regions. Line drawings are from [Lindsey 1978].*

3.1 Swimming Modes

In the field of fish physiology, the Lindsey classification of the swimming styles of various fish to 12 types is widely known [Lindsey 1978]. We call these 12 types of swimming styles "swimming modes." The classification chart for swimming modes is shown in Figure 2. In Figure 2, the vertical axis represents the types of primary movements, and the horizontal axis represents the breadth of the primary movements' regions, expressing the differences in characteristics of the swimming motions of various fish in a way that is easy to understand.

For example, Anguilliform fishes have large undulating region, spanning from body trunk to the caudal fin. On the other hand, Gymnotiform fishes undulate the long belt-shaped anal fin.

3.2 Swimming Forms

Up to this point, we have mentioned the cases where fish change their style of swimming based on the situation or speed. We call swimming styles that can change according to the situation "swimming forms." As shown in Figure 3, a number of "swimming forms" are contained in the "swimming modes."

In the proposed method, the swimming modes are treated as attributes specific to each fish species, whereas the swimming forms are treated as states that change over time. For example, a virtual fish with the Labriform swimming mode will always choose from the following three types of swimming forms: Basic-Labriform, Subcarangiform, and C-start. Our virtual fish swim by either switching their swimming form or maintaining the same swimming form over time.

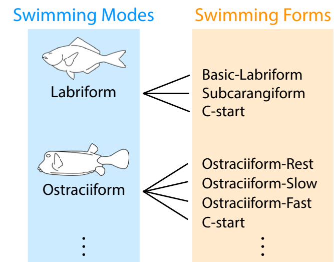

Figure 3: *Relation between swimming modes and swimming forms. For definitions of all swimming forms, refer to the supplemental document.*

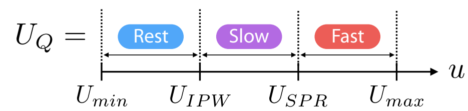

Figure 4: *Correspondence between fish speed u and qualitative speed U_Q.*

3.3 Speeds and Switching Muscles

Many fish swim using two types of muscles, red and white, depending upon the required intensity of an action [Rayner and Keenan 1967; Hudson 1973; Bone et al. 1978; Tsukamoto 1984a; Tsukamoto 1984b]. In general, red muscles are small and well-suited for sustained movement. White muscles are large and well-suited for quick movement.

Using the above as a reference, we define the qualitative speed U_Q as (Figure 4) where U_{min}, U_{IPW}, U_{SPR}, and U_{max} are species-specific parameters.

$$U_Q = \begin{cases} \text{[Rest]} & \text{if} \quad U_{min} \leq u < U_{IPW} \\ \text{[Slow]} & \text{if} \quad U_{IPW} \leq u < U_{SPR} \\ \text{[Fast]} & \text{if} \quad U_{SPR} \leq u \leq U_{max} \end{cases} \tag{1}$$

- U_{min} is the minimum speed required for breathing or generating dynamic lift.
- U_{IPW} is a parameter to represent the speed at which the white muscles start to become active.
- U_{SPR} signifies the maximum speed at which red muscles are the main muscles used and a fish can swim for a long period. U_{SPR} is also an indicator of the swimming ability of a fish.
- U_{max} represents the maximum possible speed of a fish species.

U_Q is used in the swimming form selection that is explained below. U_{min}, U_{IPW}, U_{SPR}, and U_{max} are the boundary markers used for converting u to U_Q and have the significance of being parameters that determine the motion characteristics of a fish.

4 Virtual Fish Architecture

4.1 Swimming Model

We propose a common swimming model for various fish (Figure 5) based upon the idea that fish move while instantly and repeatedly making decisions about destination, speed, and swimming style.

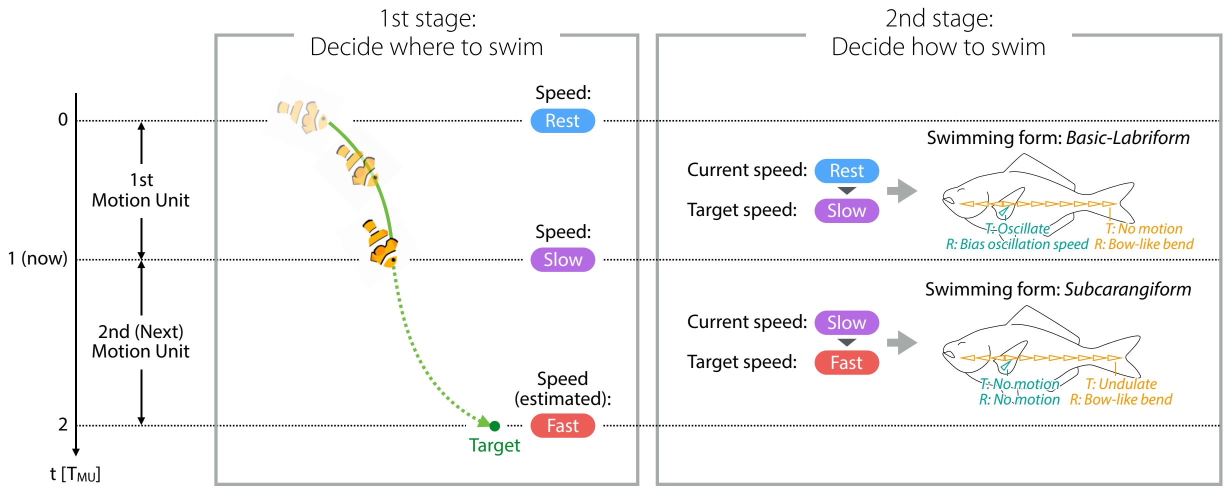

Figure 5: *Swimming model for a fish species. After each T_{MU}, the virtual fish renews its target and swimming form.*

In the proposed swimming model, fish swim by repeatedly conducting a set series of actions called Motion Units (MUs). The time required to conduct one MU is represented by T_{MU}. T_{MU} is a constant and does not change dynamically. An MU includes the following processes.

1. Deciding a target destination (target). The target represents the coordinates a fish aims to reach after time T_{MU} has elapsed.
2. Immediately after, the swimming form is decided upon using the information for the transition from the present U_Q to the target U_Q.
3. Until T_{MU} elapses, swimming is conducted in the designated swimming form, and the fish moves toward the target.

In addition, T_{MU}, which is the time length of MU, is also a parameter that can be used to describe the swimming characteristics of a fish species. For example, whale sharks, who change their movements slowly, have long T_{MU}, whereas clownfish, who move around in small detailed movements, have short T_{MU}.

4.2 Architecture Overview

The proposed architectural outline for virtual fish to implement the proposed swimming model is shown in Figure 6.

In addition, the parameters for the virtual fish, such as swimming modes, speed characteristics (U_{min}, U_{IPW}, U_{SPR}, and U_{max}), and T_{MU}, are set in advance. The Sense Association Module comprises the following four steps.

1. It senses and then inputs nearby fish, obstacles, and swimming regions or movement routes designated by the user.
2. It receives somestic sense feedback (posture, muscle fatigue) from the Locomotion Controller.
3. Based on the speed information in the feedback from the Locomotion Controller, a prediction is made on the future position of the fish after time T_{MU} has elapsed.
4. The above information is sent to the Position and Velocity Control Module, which is explained next.

The unified motion planner, which is the core of our architecture, comprises two modules: the Position and Velocity Control Module and the Swimming Form Selection Module.

1. In the Position and Velocity Control Module, the motion state of the virtual fish is randomly selected to either *active* or *inactive*. If *active* is selected, then a decision is made on schooling, escaping, slowing down (owing to fatigue), etc. based on the input information. Using the decision information as well as the estimated future position, the selection possibility distribution for the target is made and the target is decided. If *inactive* is selected, the target is decided so that the fish naturally decelerates.
2. In the Swimming Form Selection Module, the information for the transition from the present U_Q to the target U_Q is used to decide the swimming form. The swimming form selection rules of 12 swimming modes are described in the supplemental document.

Finally, the Skeleton Controller and Locomotion Controller are used to generate motion.

- The Skeleton Controller oscillates or undulates the joints of the virtual fish. The control rules differ depending upon the swimming form that is chosen.
- In the Locomotion Controller, the total position and posture of the virtual fish is controlled. The end moment of the present MU controls the acceleration and angular acceleration of the virtual fish for reaching the next target.

Details about the Position and Velocity Control Module and Locomotion Controller are given in chapter 5. Details about the Swimming Form Selection Module and Skeleton Controller are given in chapter 6.

5 Position and Velocity Control

In this section, we explain the method by which a virtual fish decides where to swim to, that is, the target selection method. At the end of this section, we also explain the details of locomotion control.

The target selection is always executed at the first time of each Motion Unit. The overall flow of the target selection is as follows:

1. A local spherical coordinate system is defined with the current position $\boldsymbol{p}_t$ of the virtual fish set as the origin (Figure 7).

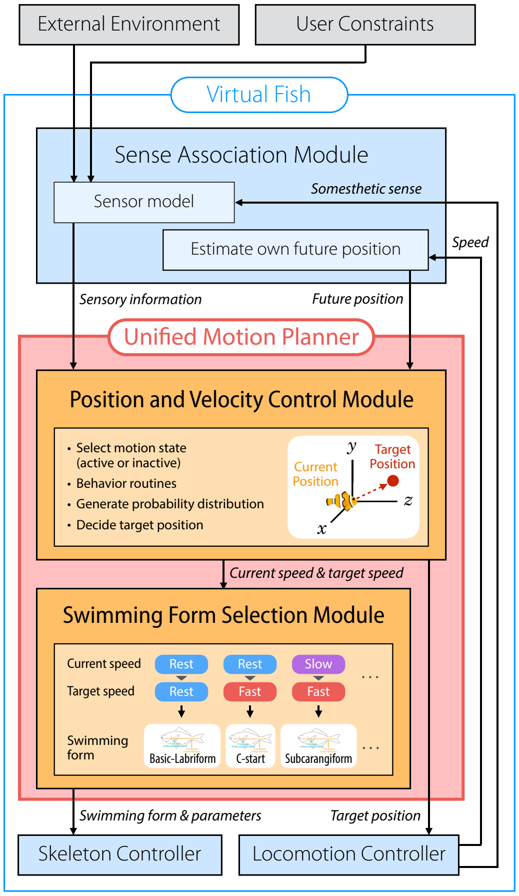

Figure 6: *Architectural outline for a virtual fish.*

2. The estimated future position $\boldsymbol{p}_{t+1}$ after T_{MU} [s] is calculated.

3. The *active* or *inactive* motion state is selected.

   - If *active* is selected, then the initial domains for two probability distributions are calculated to select the target $\boldsymbol{p}'_{t+1}$ using $\boldsymbol{p}_{t+1}$ and the parameters of the movement characteristics. The domain is clipped using sensory information and behavior routines. The probability distribution of the target is created using the final domain, and from this, target $\boldsymbol{p}'_{t+1}$ is randomly decided.

   - If *inactive* is selected, natural deceleration owing to water resistance is artificially performed. $\boldsymbol{p}'_{t+1}$ is taken as the estimated future position when natural deceleration occurs within time T_{MU} [s] from the present position $\boldsymbol{p}_t$ and present speed $\boldsymbol{u}$.

5.1 Future Position Estimation

The estimated arrival position $\boldsymbol{p}_{t+1}$ is defined as the position that will be reached if the present speed is maintained throughout T_{MU} [s] in the next MU.

Taking $\boldsymbol{p}_t$ as the present position, $\boldsymbol{q}_t$ as the present posture, and $\boldsymbol{u}$ as the present speed, $\boldsymbol{p}_{t+1}$ can be calculated with the following equation.

$$\boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \boldsymbol{q}_t \cdot |\boldsymbol{u}| \cdot T_{MU} \tag{2}$$

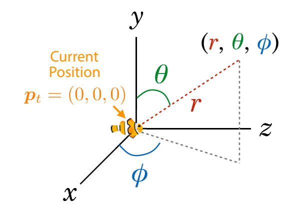

Figure 7: *The local spherical coordinate system used in the target selection. In a spherical coordinate system, the position and vectors are expressed using the radius vector r, the polar angle θ, and azimuth ϕ. We use a left-handed coordinate system for the orthogonal coordinate system.*

5.2 States and Behavior Routines

The virtual fish have the following possible state variables. S_M is related to whether they actively move and S_B is related to behavior such as escape action. S_M is called the "motion state," and S_B is the "behavior state." Both states are renewed when switching to a new MU.

S_M is either *active* or *inactive*. In other words, the state space of "motion state" is $\Omega_M = \{active, inactive\}$. S_M is chosen randomly. If $\mathrm{P_a} = \mathrm{Pr}(S_M = active)$ is the probability of selecting *active*, then the probability of selecting *inactive* is $\mathrm{P_{ia}} = 1 - \mathrm{P_a}$. $\mathrm{P_a}$ is the active rate and signifies the parameter determining how actively a virtual fish moves.

S_B is the state of *escape*, *avoid*, or *free*. In other words, the state space of the "behavior state" is $\Omega_B = \{escape, avoid, free\}$. *Escape* is the state of escaping a predator, *avoid* is the state of attempting to avoid an obstacle, and *free* is the state of swimming freely without performing either of the first two states.

S_B is selected based on the following rules.

- During the time a virtual fish is inside the set range of a predator, $S_B = escape$.
- If there is an obstacle that is neither a predator nor an individual of the same species within a set range directly in front of the virtual fish and $S_B = escape$ is not the case, then $S_B = avoid$.
- When neither of the above apply, $S_B = free$.

5.3 Probabilistic Target Selection

In the case of $S_M = active$, then based on the $\boldsymbol{p}_{t+1}$ obtained in the previous section, the domain of the probability distribution used for target selection is dynamically generated by the following procedure. Figure 8 shows an example of this process.

5.3.1 Initialize Domain of Probability Distribution

First, a local spherical coordinate system with $\boldsymbol{p}_t$ as the origin is established. Figure 8 (a) shows a view from directly above the local spherical coordinate system from the y-axis direction of an orthogonal coordinate system. The blue-lined triangle denotes the current position $\boldsymbol{p}_t$ of the virtual fish, and the cyan-blue-lined triangle marks the estimated future position $\boldsymbol{p}_{t+1}$, after T_{MU} [s].

Next, using the muscle properties, the initial domain for the probability distribution of the target is calculated (Figure 8 (b)). From the local spherical coordinate system, the probability distribution is a 3D Gaussian distribution with radius vector r, polar angle θ,

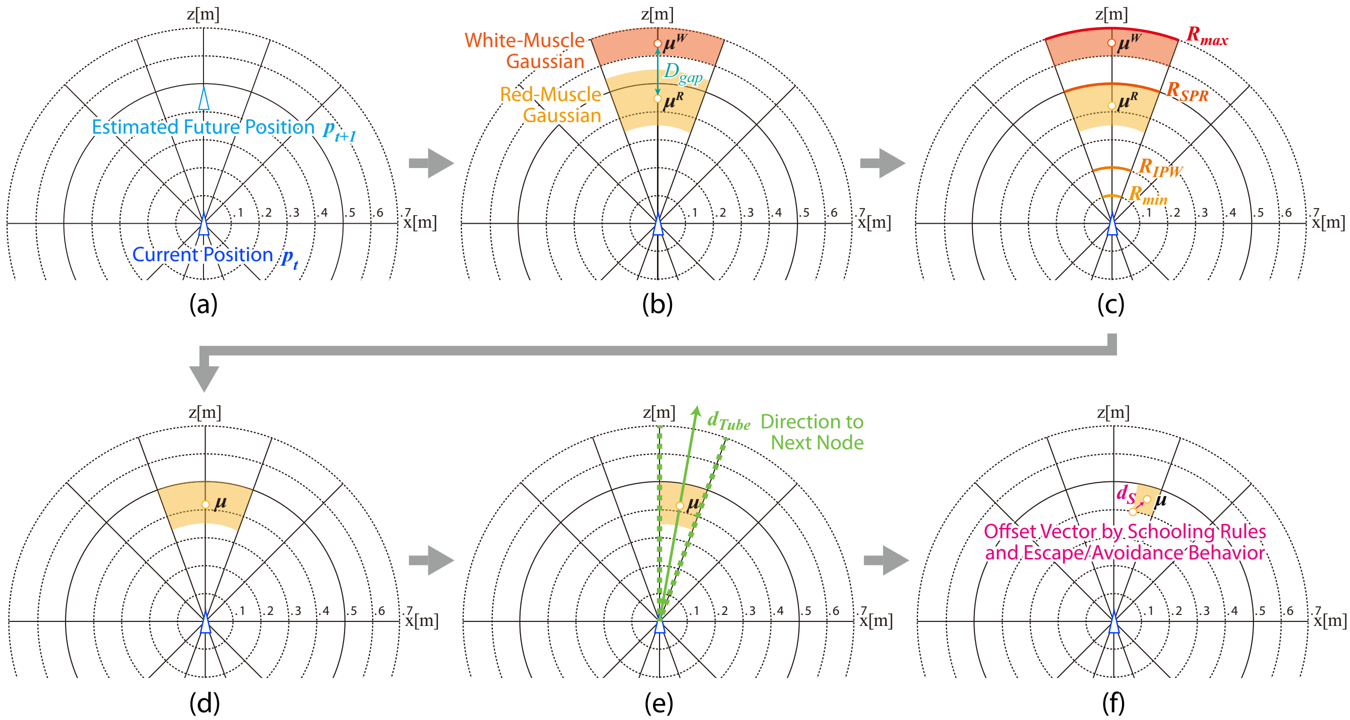

Figure 8: *An example of the process of creating domains for the target's probability distribution. (a) Top view of the local coordinate system with the virtual fish's current position as the origin. Position $\mathbf{p_t}$ is the origin, and in front of it in the z-axis direction is the estimated future position $\mathbf{p_{t+1}}$. (b) Red muscles were used in the previous MU; thus, $\boldsymbol{\mu}^{\mathbf{R}} = \mathbf{p_{t+1}}$. The position at the distance D_{gap} in the z-axis direction is $\boldsymbol{\mu}^{\mathbf{W}}$. Next, in response to the muscle characteristic parameters of the virtual fish, the initial domains $\mathbf{D^R}$ and $\mathbf{D^W}$ for the probability distributions are created around $\boldsymbol{\mu}^{\mathbf{R}}$ and $\boldsymbol{\mu}^{\mathbf{W}}$. (c) Limits based on the speed characteristics are added. R_{max}, R_{SPR}, and R_{min}, where the boundary markers for U_Q are converted to distances, are used to clip the domains. (d) Limits are imposed based on behavior routines as well as muscle fatigue. In this example, escape or the avoidance of action does not occur, and consequently, there is no muscle fatigue either. In cases like this, it is decided that the fish does not significantly accelerate the deceleration and continues to use red muscles; thus, only the RMG is considered. (e) Limits from tube following are added. The direction vector $\mathbf{d_{Tube}}$ for tracking the tube course is calculated, and the domain is clipped to within the set angle from $\mathbf{d_{Tube}}$. (f) Corrections relating to group action are made. $\boldsymbol{\mu}$ is shifted using the correction vector $\mathbf{d_S}$ from Boids [Reynolds 1987], escape and avoidance behavior, then the domain is clipped in order to place $\boldsymbol{\mu}$ at the center of the domain. A probability distribution is created using the final domain and the target $\mathbf{p'_{t+1}}$ in the next MU is probabilistically determined.*

and azimuth ϕ, as independent, continuous random variables. If $\mathbf{D}$ expresses the domain, then the probability density function is expressed as follows:

$$\Pr(r, \theta, \phi \in \mathbf{D}) = \int_{\mathbf{D}} f_{r,\theta,\phi}(r, \theta, \phi) dr d\theta d\phi \tag{3}$$

In general, in Gaussian distribution, the domain of the random variables is $\mathbf{R} = (-\infty, \infty)$, but for the target probability distribution considered in this study, the distribution is limited to within $\pm 3\sigma$ so that the distribution does not spread outside this range. First, using clipping processing, which is explained below, the domain for random variables r, θ, and ϕ and distribution mean $\boldsymbol{\mu}$ is decided. Second, a pseudo-3D Gaussian distribution with a finite domain is generated. Finally, third, using the generated probability distribution, the target is decided.

Furthermore, the initial domain of the probability distribution generates (1) red-muscle Gaussian (RMG), indicating the range a fish can move when primarily using red muscles in T_{MU}[s], and (2) white-muscle Gaussian (WMG), indicating the range a fish can move when primarily using white muscles in T_{MU}[s].

Next, $\boldsymbol{\mu}^{\boldsymbol{R}}$, the mean of RMG, and $\boldsymbol{\mu}^{\boldsymbol{W}}$, the mean of WMG, are determined. If the movement performed in the previous MU primarily used red muscles, then the estimated future position $\boldsymbol{p}_{t+1}$ is $\boldsymbol{\mu}^{\boldsymbol{R}}$ and the coordinates shifted to the set distance D_{gap} in the $+r$ direction are $\boldsymbol{\mu}^{\boldsymbol{W}}$.

$$\boldsymbol{\mu}^{\boldsymbol{R}} = \boldsymbol{p}_{t+1} = \begin{pmatrix} r_{t+1} & \theta_{t+1} & \phi_{t+1} \end{pmatrix}^{\mathrm{T}} \tag{4}$$

$$\boldsymbol{\mu}^{\boldsymbol{W}} = \begin{pmatrix} r_{t+1} + D_{gap} & \theta_{t+1} & \phi_{t+1} \end{pmatrix}^{\mathrm{T}} \tag{5}$$

On the other hand, if the movement performed in the previous MU primarily used white muscles, then the estimated future position $\boldsymbol{p}_{t+1}$ is $\boldsymbol{\mu}^{\boldsymbol{W}}$ and the coordinates shifted to the set distance D_{gap} in the $-r$ direction are $\boldsymbol{\mu}^{\boldsymbol{R}}$.

$$\boldsymbol{\mu}^{\boldsymbol{R}} = \begin{pmatrix} r_{t+1} - D_{gap} & \theta_{t+1} & \phi_{t+1} \end{pmatrix}^{\mathrm{T}} \tag{6}$$

$$\boldsymbol{\mu}^{\boldsymbol{W}} = \boldsymbol{p}_{t+1} = \begin{pmatrix} r_{t+1} & \theta_{t+1} & \phi_{t+1} \end{pmatrix}^{\mathrm{T}} \tag{7}$$

Finally, the scope of the two probability distributions, in other words, the random-variable domains $\mathbf{D^R}$ and $\mathbf{D^W}$ for the respective distribution, is determined.

If W_r^R is the extent of the distribution relating to r in the RMG, W_r^W is the extent of the distribution with respect to r in the WMG, W_θ is the extent of the distribution with respect to ϕ, and W_ϕ is the extent of the distribution with respect to ϕ, then

$$W_r^R = 2\ddot{r}_{max}^R \cdot T_{MU}^2 \quad (8)$$

$$W_r^W = 2\ddot{r}_{max}^W \cdot T_{MU}^2 \quad (9)$$

$$W_\theta = 2\ddot{\theta}_{max} \cdot T_{MU}^2 \cdot \frac{r_{t+1}}{R_{max} - R_{min}} \quad (10)$$

$$W_\phi = 2\ddot{\phi}_{max} \cdot T_{MU}^2 \cdot \frac{r_{t+1}}{R_{max} - R_{min}} \quad (11)$$

Here, $\ddot{\theta}_{max}$ is the maximum angular acceleration in the θ direction, $\ddot{\phi}_{max}$ is the maximum angular acceleration in the ϕ direction, $\ddot{r}_{max}^R$ is the maximum angular acceleration of the red muscles, and $\ddot{r}_{max}^W$ is the maximum angular acceleration of the white muscles. These are parameters for the muscle characteristics of a virtual fish; moreover, $R_{max} = U_{max} \cdot T_{MU}$ and $R_{min} = U_{min} \cdot T_{MU}$. R_{max} and R_{min} correspond to the maximum and minimum value of r, respectively, that the virtual fish can travel in the next MU. In addition, we expand the width of θ and ϕ of the probability distributions relative to r_{t+1}, which is the r component for the estimated future position, to avoid cases where the fish turn too much when moving slightly forward.

Four random variables r^R, r^W, θ, and ϕ are used. The domains for these random variables, $\mathbf{D_r^R}$, $\mathbf{D_r^W}$, $\mathbf{D}_\theta$, and $\mathbf{D}_\phi$ can be obtained by the following equations.

$$\mathbf{D_r^R} = \left\{ r^R \mid \mu_r^R - \frac{W_r^R}{2} \le \mu_r^R \le \mu_r^R + \frac{W_r^R}{2} \right\} \quad (12)$$

$$\mathbf{D_r^W} = \left\{ r^W \mid \mu_r^W - \frac{W_r^W}{2} \le \mu_r^W \le \mu_r^W + \frac{W_r^W}{2} \right\} \quad (13)$$

$$\mathbf{D}_\theta = \left\{ \theta \mid -\frac{W_\theta}{2} \le 0 \le \frac{W_\theta}{2} \right\} \quad (14)$$

$$\mathbf{D}_\phi = \left\{ \phi \mid -\frac{W_\phi}{2} \le 0 \le \frac{W_\phi}{2} \right\} \quad (15)$$

Here, μ_r^R is the r component of $\boldsymbol{\mu}^{\boldsymbol{R}}$ and μ_r^W is the r component of $\boldsymbol{\mu}^{\boldsymbol{W}}$.

Using the above, the domain $\mathbf{D^R}$ of RMG and the domain $\mathbf{D^W}$ of WMG are defined using the following equations, respectively.

$$\mathbf{D^R} = \mathbf{D_r^R} \times \mathbf{D}_\theta \times \mathbf{D}_\phi \quad (16)$$

$$\mathbf{D^W} = \mathbf{D_r^W} \times \mathbf{D}_\theta \times \mathbf{D}_\phi \quad (17)$$

5.3.2 Constraint by Speed Features

Next, the following three limitations based on the speed characteristic parameters are added to domains $\mathbf{D^R}$ and $\mathbf{D^W}$.

1. A fish cannot swim faster than U_{SPR} when primarily using its red muscles.
2. When a fish is moving primarily using its white muscles, it can swim faster than U_{SPR} but it cannot swim faster than U_{max}.
3. A fish cannot swim slower than U_{min}.

Next, we convert U_{max}, U_{SPR}, and U_{min} into distance values, as shown below.

$$R_{max} = U_{max} \cdot T_{MU} \quad (18)$$

$$R_{SPR} = U_{SPR} \cdot T_{MU} \quad (19)$$

$$R_{min} = U_{min} \cdot T_{MU} \quad (20)$$

Clipping of the domain is achieved by calculating the intersection of a closed interval with the values above as end points and using the domains $\mathbf{D_r^R}$ and $\mathbf{D_r^W}$ of the probability distribution (Figure 8 (c)).

$$\mathbf{D_r^R} \leftarrow \mathbf{D_r^R} \cap [R_{min}, R_{SPR}] \quad (21)$$

$$\mathbf{D_r^W} \leftarrow \mathbf{D_r^W} \cap [R_{min}, R_{max}] \quad (22)$$

When clipping the domains, the mean $\boldsymbol{\mu}$ of the probability distribution is updated by substituting with the mean of the end points of the closed interval representing the domain. Subsequent clipping is calculated in a similar manner.

5.3.3 Constraint by Behavior Routines

Next, constraints are added based on the virtual fish's behavior state S_B.

If $S_B = escape$, then white-muscle action is performed for acceleration. With $\max(r^W)$ representing the maximum value of $\mathbf{D_r^W}$, the following equation is used to clip $\mathbf{D_r^R}$.

$$\mathbf{D_r^W} \leftarrow \mathbf{D_r^W} \cap \left[\mu_r^W, \max(r^W) \right] \quad (23)$$

In addition, because the red muscles are not used, RMG is not be used.

In the case of $S_B = avoid$ or $S_B = free$, red-muscle action is performed. Due to the fact that the white muscles are not used, WMG is not used (Figure 8 (d)).

At this point, a decision is made on whether to use WMG or RMG. Thus, hereafter the domain of the probability distribution is denoted as $\mathbf{D}$, and the domain of the random variable r is denoted as $\mathbf{D_r}$.

5.3.4 Constraint by Muscle Fatigue

Next, constraints owing to muscle fatigue are added.

When the virtual fish speed u exceeds U_{SPR}, oxygen debt O_D [Gaesser and Brooks 1984] accumulates at a rate that is proportional to the square of the difference between the present speed and U_{SPR} speed. Using Δt, which is the elapsed time between frames, we simply integrate.

$$O_D \leftarrow O_D + (u - U_{SPR})^2 \cdot \Delta t \quad (24)$$

When O_D reaches its limit value $\max(O_D)$, continuing to swim at the present speed becomes impossible, and a constraint is added where the virtual fish is forced to decelerate and reduce its O_D value. Thus, if $\min(r)$ is the minimum value of $\mathbf{D_r}$, the following equation is used to clip $\mathbf{D_r}$.

$$\mathbf{D_r} \leftarrow \mathbf{D_r} \cap [\min(r), \mu_r] \quad (25)$$

When O_D falls below U_{SPR}, O_D decreases at the set rate. Once O_D reaches 0, the constraint is removed.

5.3.5 Tube-Following

Next, constraints are added based on the user's path designation.

Path-Following is widely used to establish the movement path; however, in this study, in order for users to easily designate the motion of large schools of fish, we use Tube-Following, wherein the path of movement of virtual fish is designated as the tube course.

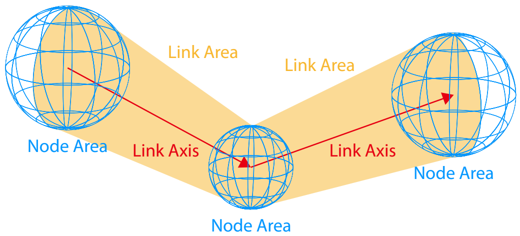

Figure 9: *An example of a tube course construction. The blue sphere is the Node Area, the orange region connecting each Node Area is the Link Area, and the red line that passes through the center of the Link Area is the Link Axis.*

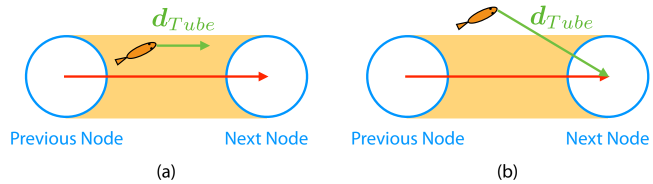

Figure 10: *Method for calculating the direction vector $\mathbf{d}_{Tube}$, which is the standard for the constraint. Top view of the tube course that runs from left to right. (a) If $\mathbf{p}_t$ lies inside the Link Area, the Link Axis vector presently being tracked is taken as $\mathbf{d}_{Tube}$. (b) If $\mathbf{p}_t$ lies outside the Link Area, the vector toward the Node Area that is being tracked is taken as $\mathbf{d}_{Tube}$.*

The tube course is obtained by creating a number of spherical regions (Node Areas), assigning an order to the node areas and then connecting them with lines. The conical area formed by the node areas is called the "Link Area," and the vector constituting the axis of the Link Area is the Link Axis. We show a tube course comprising three Node Areas in Figure 9, as an example.

A virtual fish performing Tube-Following will constantly have a specific Node Area as well a Link Area leading to this Node Area and the targets it is tracking. The moment a fish reaches the inside of the targeted Node Area, the tracking is switched to target the next Node and Link Area(s).

In addition, to make the fish swim in accordance with the tube course, the domain $\mathbf{D}_\theta$ of θ and the domain $\mathbf{D}_\phi$ of ϕ of the probability distributions are clipped to make them fall within the set angle around the limit vector $\boldsymbol{d_{Tube}}$ (Figure 8 (e)).

$$\mathbf{D}_\theta \leftarrow \mathbf{D}_\theta \cap \left[\theta_{Tube} - \frac{W_{\theta_{Tube}}}{2}, \theta_{Tube} + \frac{W_{\theta_{Tube}}}{2}\right] \tag{26}$$

$$\mathbf{D}_\phi \leftarrow \mathbf{D}_\phi \cap \left[\phi_{Tube} - \frac{W_{\phi_{Tube}}}{2}, \phi_{Tube} + \frac{W_{\phi_{Tube}}}{2}\right] \tag{27}$$

Here, θ_{Tube} and ϕ_{Tube} are the angles that result from converting the $\boldsymbol{d_{Tube}}$ in terms of the θ and ϕ of the local spherical coordinate system of the virtual fish, respectively. $W_{\theta_{Tube}}$ and $W_{\phi_{Tube}}$ are parameters indicating the size of the angle being limited and are designated upon the tube course. When these values are large, the constraints are gentle with some random scattering occurring in the movement of the fish schools. When these values are small, coherent line-like movement occurs.

As for the method of calculating $\boldsymbol{d_{Tube}}$, depending on whether the present position $\boldsymbol{p_t}$ of the virtual fish lies inside or outside the Link Area, one of the following is selected.

- If $\boldsymbol{p_t}$ lies inside the Link Area, the Link Axis vector presently being tracked is directly taken to be $\boldsymbol{d_{Tube}}$. (Figure 10 (a))
- If $\boldsymbol{p_t}$ lies outside Link Area, the vector directed at the Node Area currently being tracked is taken as $\boldsymbol{d_{Tube}}$ (Figure 10 (b)).

In addition, the tube course can be made to create circling movement. If the Node Area size is set at 0, the fish will track only the Link Axis; thus, in this manner, the tube course can also be used for "Path Following." A tube course with only one Node Area and without any Link Areas can be used to keep a virtual fish within a set range, such as a fish tank.

5.3.6 Adjust by Schooling Behavior

Finally, a constraint based on schooling behavior is added only when the virtual fish act as a school. Furthermore, in the cases of $S_B = escape$ or $S_B = avoid$, constraints for moving in the opposite direction of the object-to-avoid, such as a predator, are also added.

For schooling action, we use Boids [Reynolds 1987], a widely used method of crowd simulation. Using the three rules from Boids (cohesion rule, alignment rule, and separation rule), the acceleration vector $\boldsymbol{a_B}$ that is used to correct the movement of each individual fish is calculated.

The acceleration vector $\boldsymbol{a_E}$ for moving in the opposite direction of the object-to-avoid is calculated using the following equation.

$$\boldsymbol{a_E} = \begin{cases} \frac{\boldsymbol{d_o}}{||\boldsymbol{d_o}||} K_E (1 - \frac{||\boldsymbol{d_o}||}{D_{safety}}) & \text{if } ||\boldsymbol{d_o}|| < D_{safety} \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

Parameter $\boldsymbol{d_o}$ is the vector from the object-to-avoid to the virtual fish, K_E expresses the degree of avoidance, and D_{safety} denotes the distance that is the threshold for performing the act of avoidance. When the distance between the virtual fish and object-to-avoid becomes less than D_{safety}, the acceleration $\boldsymbol{a_E}$, which is inversely proportional to the distance, is generated.

Using $\boldsymbol{a_B}$ and $\boldsymbol{a_E}$, the correction vector $\boldsymbol{d_S}$, which is used for clipping the domain of the probability distribution, is obtained using the following equations.

$$\boldsymbol{d_B} = \boldsymbol{a_B} \cdot T_{MU}^2 \tag{29}$$

$$\boldsymbol{d_E} = \boldsymbol{a_E} \cdot T_{MU}^2 \tag{30}$$

$$\boldsymbol{d_S} = L_S \boldsymbol{d_B} + (1 - L_S) \boldsymbol{d_E} \tag{31}$$

Parameter L_S is for determining the extent to which schooling action is prioritized over avoidance action ($0 \leq L_S \leq 1$).

To correct $\boldsymbol{\mu}$, $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \boldsymbol{d_S}$ is used, and $\mathbf{D}$ is clipped to make the corrected $\boldsymbol{\mu}$ the center of domain $\mathbf{D}$ (Figure 8 (f)).

From the above procedure, domain $\mathbf{D}$ and mean $\boldsymbol{\mu}$, which satisfy all given constraints, are obtained and the 3D Gaussian distribution is created. Using the Box-Muller transform [Box and Muller 1958], each component of the target coordinates $\boldsymbol{p}'_{t+1} = \begin{pmatrix} r'_{t+1} & \theta'_{t+1} & \phi'_{t+1} \end{pmatrix}^{\mathrm{T}}$ is determined probabilistically. If $r'_{t+1} > r_{t+1}$, the virtual fish will accelerate in the next MU, and if $r'_{t+1} < r_{t+1}$, the virtual fish will decelerate in the next MU.

5.3.7 Natural Slowdown

For $S_M = inactive$, natural deceleration owing to water resistance is conducted artificially. The estimated future position after continuing to naturally decelerate from speed $\boldsymbol{u}$ over the time period T_{MU}[s] is the target $\boldsymbol{p}'_{t+1}$.

$$\boldsymbol{a_D} = -\frac{1}{2m}\rho \begin{pmatrix} u_x^2 \\ u_y^2 \\ u_z^2 \end{pmatrix} SC_D \tag{32}$$

$$\boldsymbol{p}'_{t+1} = (\boldsymbol{a_D} \cdot T_{MU} + \boldsymbol{u}) \cdot T_{MU} \tag{33}$$

Here, $\boldsymbol{a_D}$ is the acceleration owing to water resistance. In addition, m is the mass of the virtual fish, ρ is the water density, S is the representative surface area of the virtual fish, and C_D is the drag coefficient (coefficient of resistance). All these parameters are constant.

5.4 Locomotion Control

The Locomotion Controller controls the action of the entire body and, consequently, the fish arrive at the target exactly when time T_{MU}[s] has elapsed.

Regarding translational motion, the acceleration a'_z in the z-axis direction allows for the exact arrival at $\boldsymbol{p'_{t+1}}$ if uniformly accelerated motion is conducted until MU is complete, calculated at every frame, and speed $\boldsymbol{u}$ is updated.

$$a'_z = 2\left(\frac{d_z}{t_r^2} - \frac{u_z}{t_r}\right) \tag{34}$$

$$u_z \leftarrow u_z + a'_z \cdot \Delta t \tag{35}$$

Parameter d_z represents the distance in the z-axis direction from the present position $\boldsymbol{p_t}$ to target $\boldsymbol{p'_{t+1}}$ in the local coordinate system of the virtual fish. In addition, t_r is the remaining time of MU and Δt is the time elapsed between frames.

Regarding rotational motion, the unit vector $\boldsymbol{q_{target}}$ pointing from $\boldsymbol{p_t}$ to $\boldsymbol{p'_{t+1}}$ is established as the target posture. The PID control is used to obtain the angular velocity $\boldsymbol{\alpha}'$ that causes the virtual fish posture to approach that of the target posture, and the process of updating the angular speed $\boldsymbol{\omega}$ is executed for each frame.

$$\Sigma\boldsymbol{q_e} \leftarrow \Sigma\boldsymbol{q_e} + \boldsymbol{q_e} \tag{36}$$

$$\boldsymbol{\alpha}' = K_p\boldsymbol{q_e} + K_i\Sigma\boldsymbol{q_e} + K_d\frac{\boldsymbol{q_e} - \boldsymbol{q_{ep}}}{\Delta t} \tag{37}$$

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} + \boldsymbol{\alpha}' \cdot \Delta t \tag{38}$$

$\boldsymbol{q_e}$ is the relative angle between the posture of the virtual fish in the present frame and $\boldsymbol{q_{target}}$, $\Sigma\boldsymbol{q_e}$ is the integrated value of $\boldsymbol{q_e}$, and $\boldsymbol{q_{pe}}$ is the $\boldsymbol{q_e}$ value of the previous frame. Constants K_p, K_i, and K_d correspond to the P gain, I gain, and D gain, respectively.

6 Swimming Form Selection

In this section, the specific definition of the swimming form, the selection method of the swimming form, and the skeleton control method are explained in detail.

6.1 Partial Skeleton Model

Fish species, starting with their body trunk, have many movable parts, such as pectoral fins, caudal fins, dorsal fins, and anal fins, but the parts that move significantly when swimming are few. In addition, many parts have similar structures and ways of being moved. Thus, from the perspective of skeletal structure and how this skeletal structure is moved, we divide the major body parts that move when swimming into four types of Partial Skeleton Units (PSUs).

1. Body Trunk - Caudal Fin Unit (Body-PSU): Corresponds to the part from the body trunk to the caudal fin. Like the spine, it has a skeletal structure that connects in a single-row series.

2. Plate-like Fin Unit (Plate-PSU): A plate- and fin-like small pectoral fin.

3. Ribbon-like Fin Unit (Ribbon-PSU): A long ribbon-shaped fin, observed in the dorsal fin of Amiiform fish, the caudal fin of Gymnotiform fish, etc. A series of horizontal rays move in a coordinated manner.

4. Disk-type Pectoral Fin Unit (Disk-PSU): A giant pectoral fin that also connects to the head region is called a disk. It is a characteristic of Rajiform fish and is not observed outside of Rajiform fish.

In our method, we define each fish species skeleton corresponding to the 12 types of swimming modes as a partial skeleton model, which is a combination of multiple PSUs.

Moreover, the swimming form is modeled by mapping each translational motion and rotational motion, and the four types of basic movements, i.e., "oscillate," "undulate," "bow-like bend," and "no motion," for each PSU. As an example, in Figure 11, we show the construction of the partial skeleton model and swimming form definitions for Labriform and Ostraciiform fish.

The partial skeleton model for Labriform fish comprises a Body-PSU and a pair of Plate-PSUs, which correspond to the pectoral fin. When the swimming form is the Basic-Labriform, the Labriform fish propels forward by oscillating the pectoral fin ' s Plate-PSUs and changes direction by bending the Body-PSU into a bow. In addition, like the oars of a boat, by moving the pectoral fins at an uneven speed from left to right, turning is achieved in a natural manner. When the swimming form is the Subcarangiform, the Plate-PSUs do not move; instead, the Body-PSU undulates to bring about translational motion. In all swimming modes, C-start is modeled as bending the Body-PSU in order to turn.

Since boxfish, owing to their body structure, can only bend their bodies slightly, the Body-PSU for Ostraciiform is made extremely short. In addition, Plate-PSUs are allotted to the pectoral fins, dorsal fins, and anal fins. When the swimming form is the Ostraciiform-Rest, the dorsal fin and caudal fin do not oscillate and only bend in rotational motion; however, as the swimming form changes, the dorsal fin and caudal fin start to oscillate along with the pectoral fins.

6.2 Swimming Form Selection

When fish accelerate at a particular acceleration rate, sometimes even for the same acceleration rate, transitioning from a mostly still state to a slow swimming state and transitioning from a slow swimming state to a fast swimming state can be totally different. Thus, in the proposed method, we obtain the qualitative speed $\boldsymbol{U}_{Q_{t+1}}$ in the next MU and pair the swimming form and the transition information for transitioning from the qualitative speed $\boldsymbol{U}_{Q_t}$ to $\boldsymbol{U}_{Q_{t+1}}$ within the present MU. By doing so, we allow for sudden changes in the swimming form in response to changes.

$\boldsymbol{U}_{Q_{t+1}}$ is obtained by using the following formula and the r com-

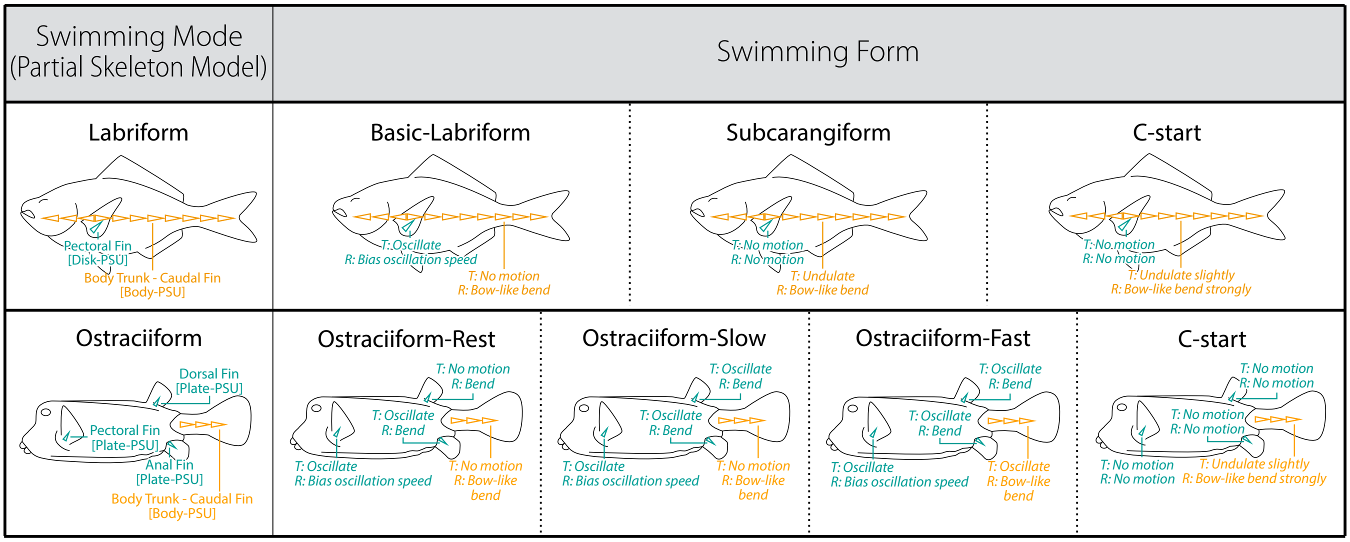

Figure 11: *The definition of partial skeleton models and swimming forms for Labriform and Ostraciiform fish. T represents the action corresponding to translational motion, and R represents the action corresponding to rotational motion; these parameters proportionally change with the speed and angular speed, respectively. Regarding the definitions for the other swimming modes, please refer to the supplemental document.*

Table 1: *Swimming form selection rules for Labriform.*

$\boldsymbol{U}_{Q_t}$	$\boldsymbol{U}_{Q_{t+1}}$	S-Form
Rest	Rest	B-Labf.
Rest	Slow	B-Labf.
Rest	Fast	C-start
Slow	Rest	B-Labf.
Slow	Slow	B-Labf.
Slow	Fast	Subcf.
Fast	Rest	B-Labf.
Fast	Slow	B-Labf.
Fast	Fast	Subcf.

Table 2: *Swimming form selection rules for Ostraciiform.*

$\boldsymbol{U}_{Q_t}$	$\boldsymbol{U}_{Q_{t+1}}$	S-Form
Rest	Rest	Ostf.-Rest
Rest	Slow	Ostf.-Slow
Rest	Fast	C-start
Slow	Rest	Ostf.-Rest
Slow	Slow	Ostf.-Slow
Slow	Fast	Ostf.-Fast
Fast	Rest	Ostf.-Rest
Fast	Slow	Ostf.-Slow
Fast	Fast	Ostf.-Fast

ponent r'_{t+1} from target $\boldsymbol{p}'_{t+1}$. Also, $R_{IPW} = U_{IPW} \cdot T_{MU}$.

$$\boldsymbol{U}_{Q_{t+1}} = \begin{cases} [\text{Rest}] & \text{if} \quad R_{min} \leq r'_{t+1} < R_{IPW} \\ [\text{Slow}] & \text{if} \quad R_{IPW} \leq r'_{t+1} < R_{SPR} \\ [\text{Fast}] & \text{if} \quad R_{SPR} \leq r'_{t+1} \leq R_{max} \end{cases} \tag{39}$$

For $\boldsymbol{U}_{Q_t}$, the $\boldsymbol{U}_{Q_{t+1}}$ calculated during the previous MU update is used directly. However, for the first MU, $\boldsymbol{U}_{Q_t} = $ [Rest] is used as the initial value.

Specifically, we list the swimming form selection rules for Labriform and Ostraciiform in Tables 1 and 2, respectively. Here, S-form is the Swimming Form, B-labf. is the Basic-Labriform, Subcf. is the Subcarangiform, and Ostf. is the Ostraciiform. Regarding the swimming form selection rules for the other swimming modes, please refer to the supplemental document.

For all swimming modes, whenever transitioning from rest to fast and escape action, the C-start is selected. For Labriform, Subcarangiform is selected whenever transitioning from [Slow] to [Fast] and from [Fast] to [Fast], and for other cases, Basic-Labriform is selected. Similarly, for Ostraciiform, besides C-start, Ostraciiform-Rest is selected when $\boldsymbol{U}_{Q_{t+1}} = $ [Rest], Ostraciiform-Slow is selected when $\boldsymbol{U}_{Q_{t+1}} = $ [Slow], and Ostraciiform-Fast is selected when $\boldsymbol{U}_{Q_{t+1}} = $ [Fast].

6.3 Skeleton Control

Regarding the movement of PSUs, the movement is basically a combination of oscillation, undulation, and bow-like bending.

For recreating oscillation and undulation, we use an equation from Willy et al. [Willy and Low 2005], who applied to fish robots the undulation model of an eel by Lighthill [Lighthill 1971].

$$y = Ae^{\alpha(s-1)} \sin k(s - Vt) \tag{40}$$

Where y is the displacement from wave motion, A is the amplitude parameter, α is the degree of the spread of the wave from one side to the other, k is the number of appearing waves, and V is the propagation velocity of the wave. The angle of each of the joints is bent to approximate the waveform obtained with this equation. When k is large, undulation appears, and when k is small, oscillation appears.

When fish propel forward, in addition to the force needed for accelerating, they need to move their bodies to exert the force necessary for cancelling the effect of water resistance. Thus, for calculating V, we use the resultant force from these two forces. By doing this, we actualize the movement that appears like swimming in water, while substantially simplifying the calculations.

Regarding bending, the PID control is conducted using a quantity that is exponentially reduced from $\boldsymbol{\omega}_m$, the mean movement of angular speed, as the target angle. In other words, if d is the distance from the root of the joint, the initial value of the target angle is taken as $\theta_0 = \boldsymbol{\omega}_m$ and the disintegration constant is $\lambda(\lambda \geq 0)$. Then, the joint's target angle θ is expressed by the following equation.

$$\theta(d) = \theta_0 e^{-\lambda d} \tag{41}$$

The basic movement of the plate PSU is identical to the movement of the Body-PSU with only one joint. However, since the Plate-PSUs are often used for pectoral fins, it is necessary to consider that actual fish will, in many cases, turn by making the speed of moving their pectoral fins uneven, from left to right, like the oars of a boat. Thus, by adding an offset to the V of the left and right Plate-PSUs, in connection to the angular speed $\boldsymbol{\omega}$, natural turning movement is achieved.

For a disk PSU, multiple undulating Body-PSUs are placed horizontally and side-by-side, and the movement is approximated by shifting the phase by a set quantity.

7 Results

7.1 Implementation Details

We applied the proposed method using a script from the 3D game engine Unity and C#. The script operates on a single thread. In this section, we show the results of operating our simulator on a 3.60 GHz CPU.

Our simulator has a high computational load, particularly in regard to the nearest neighbor search during crowd simulation. We decreased the calculation time for the nearest neighbor search using a kd-tree [Bentley 1975]. However, to efficiently process schooling action or avoidance action in situations often seen in underwater scenes, where fish exhibiting schooling behavior and other fish exist together, we use the following three cases as categories:

1. The nearest neighbor search that a fish performing schooling action conducts with respect to fish of the same species. To do this, the kd-tree handles only fish that perform schooling action.

2. The nearest neighbor search that a fish not participating in schooling action conducts to avoid other fish performing schooling action. To do this, kd-tree handling of all fishes in the scene is used.

3. The nearest neighbor search that a fish participating in schooling action conducts to detect predators or obstacles. In this case, the predators and obstacles, which are the objects being searched for, are few and to prioritize avoiding for the cases where not everything is detected, the kd-tree is not used and distances are calculated with respect to all objects.

We apply the proposed method to rigged CG models. There are 12 types of CG models and their shape and skeletal construction corresponds to each of the 12 types of swimming modes. The number of polygons and Degrees of Freedom (DoF) are given in Table 3. The DoF relating to the rotation differ according to PSU. For the case of Body-PSU only, undulation occurs not only in the x-axis direction but also in the y-axis direction. In other words, the Body-PSU has two DoF per joint, whereas all other PSUs have one DoF per joint.

7.2 Simulation Results

In Figure 12, we show the results of performing simulations for all 12 types of swimming modes. To see the swimming animation in detail, please refer to the supplemental video. Thus, the proposed method can be applied to various fish species with very different sizes and skeletons.

In Figure 13, we show a Labriform fish swimming while switching swimming forms. First, it swims slowly using the Basic-Labriform. When a predator approaches, the fish uses C-start for moment, significantly bending its body, and distances itself from the predator. Even after C-start has been completed, it uses Subcarangiform for a while and continues to escape quickly. The Ostraciiform fish, shown in Figure 14, is similarly able to select swimming forms for

Table 3: *Details of the CG models. Num Tri denotes the number of triangular polygons, while Num DoF denotes the total DoF for the entire skeleton.*

Swimming mode	Num Tri	Num DoF
Anguilliform	6092	30
Subcarangiform	582	24
Carangiform	7984	32
Thunniform	2024	30
Ostraciiform	11200	14
Amiiform	19776	51
Gymnotiform	10496	48
Balistiform	16000	42
Tetraodontiform	13952	28
Rajiform	2696	90
Diodontiform	6592	51
Labriform	2752	24

Table 4: *List of parameters.*

Parameter	Adult Fish	Young Fish	Fry
T_{MU}	0.5	0.1	0.06
U_{max}	6	3	0.8
U_{SPR}	2	1	0.25
U_{min}	0.15	0.075	0.03

cases of swimming slowly, swimming quickly, and swimming to escape from a predator.

In the proposed method, the movement variation is created in the same CG model simply by changing the number of parameters. In Figure 15, we show the comparison of three movement patterns created from changing the number of parameters for the same Carangiform fish. In addition, the changed parameters and set values are provided in Table 4.

In the proposed method, the variations in the shapes of schools of fish can easily be created by establishing the tube course. In Figure 16, we show the results of two types of different school-of-fish simulations for the pilchard. The pilchard's swimming mode is Subcarangiform. By simply changing the arrangement of the tube course and the population of the pilchards, the torus- or tornado-type shape, which is often observed in actual schools of fish, can be recreated.

One significant characteristic of the proposed method is the light calculation load. By doing this, as we showed in Figure 1, we are actualizing the simultaneous simulation with the simulation of various fish species as well as the simulation of schools of fish with many fish, creating a scene of an actual underwater environment. We had a school of pilchards swim in a torus-type shape, as shown in the left picture of Figure 16, and we measured the drawing frame rate while varying the number of fish. We provide the results in Table 5. Real-time operation was possible for a population of around 500 fish and, in non-real-time cases, schools of fish of 10,000 or more can be simulated.

Next, in Figure 17, we compare the shapes of schools of fish with different $W_{\theta_{Tube}}$ and $W_{\phi_{Tube}}$ values, which are the specified limits for the angle of $\boldsymbol{d_{Tube}}$, the limit vector used for tracking. When $W_{\theta_{Tube}} = W_{\phi_{Tube}} = 0°$, the tube course is tightly constrained and the shape of the school of fish is linear. For $W_{\theta_{Tube}} = W_{\phi_{Tube}} = 90°$, a moderate level of scattering occurs in the school of fish. However, for $W_{\theta_{Tube}} = W_{\phi_{Tube}} = 180°$, the number of fish that leave the tube course and cannot return increases, and thus, the cohesion of the school of fish is lost. In this manner, in the proposed method, the level of cohesion of a school

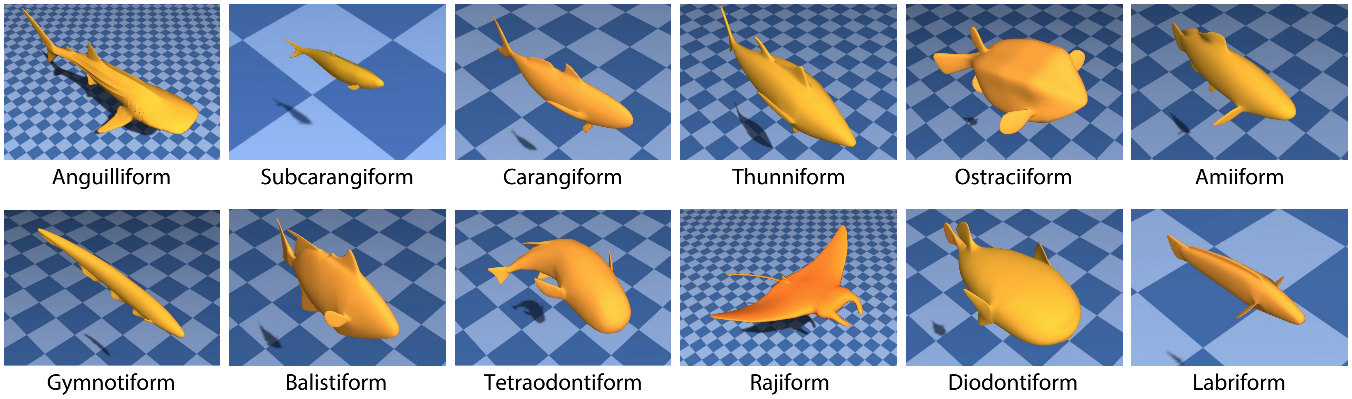

Figure 12: *Simulation results for each of the 12 types of swimming modes.*

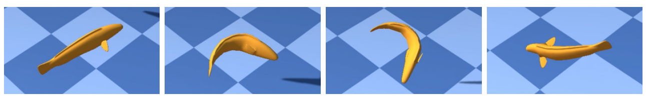

Figure 13: *A Labriform fish swims while switching swimming forms.*

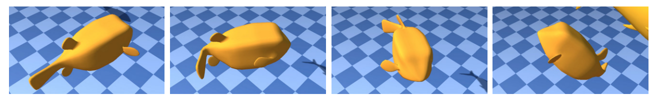

Figure 14: *An Ostraciiform fish swims while switching swimming forms.*

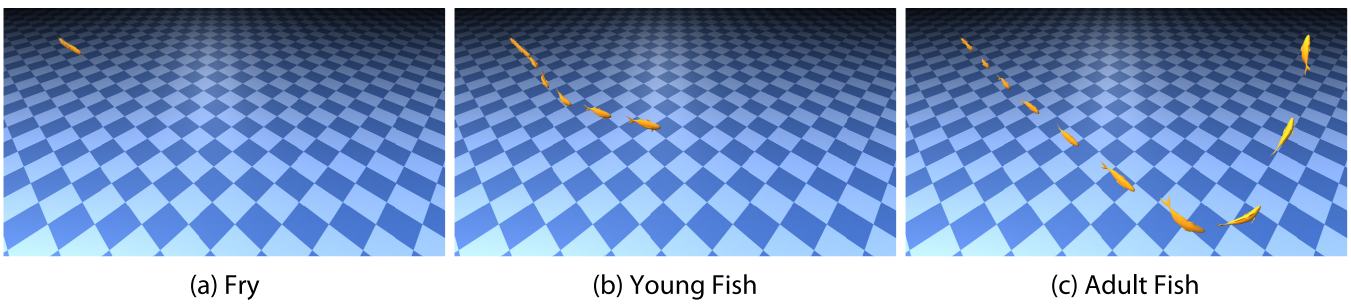

Figure 15: *An example wherein we created movement variations by changing the number of parameters for the same Carangiform fish. Three patterns are shown: swimming with a lot of short quick movements like a fry, swimming with some short quick movements like a young fish, and swimming smoothly like an adult fish.*

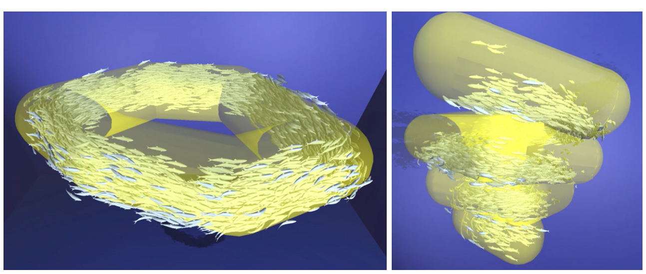

Figure 16: *School of pilchards with 4,000 fish, swimming in torus-type shape (left). School of pilchards with 8,000 fish, swimming in tornado-type shape (right). We show the tube course in a semi-transparent yellow color.*

of fish can be easily adjusted.

In Figure 18, we show an example of the dynamic shape change of a school of fish brought about by the attacking action of a predator. In this example, a tuna, in the role of the predator, enters the central area of a school of pilchards. The pilchards close to the tuna individually perform the escape action and the entire school of fish disperses at once; however, when the tuna is gone, the pilchards will gradually form a shoal and regroup to the tube course they were originally swimming.

Table 5: *Comparison of the calculation load. Num Fish denotes the number of fish. Sim Time is the calculation time in the simulation.*

Num Fish	FPS	Sim Time[ms]
100	60	16.5
500	18	82.8
1000	4	307.6
5000	0.5	1882.0
10000	0.24	3952.2
15000	0.15	6081.4

In Figure 19, we show the robustness of the motion control against an external force intended to model the water current. We evenly applied a perpendicular external force to the tube course and the virtual fish. The virtual fish are affected by the external force and try to move along the tube course. When the external force is large, it becomes difficult for the virtual fish to approach their target, and they are pushed outside of the tube course. Therefore, even in the case of an external force, we were able to realistically recreate the

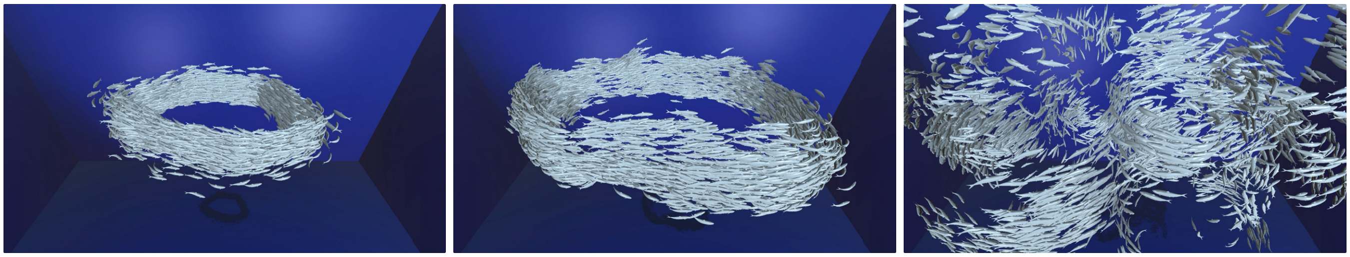

Figure 17: *Comparing parameters for a school of fish comprising 4,000 fish swimming in a torus shape. $W_{\theta_{Tube}}$ and $W_{\phi_{Tube}}$, which are the specified limits of the angles of the limit vector $\boldsymbol{d_{Tube}}$ used for tracking, are varied. Left is 0°, center is 90°, and right is 180°.*

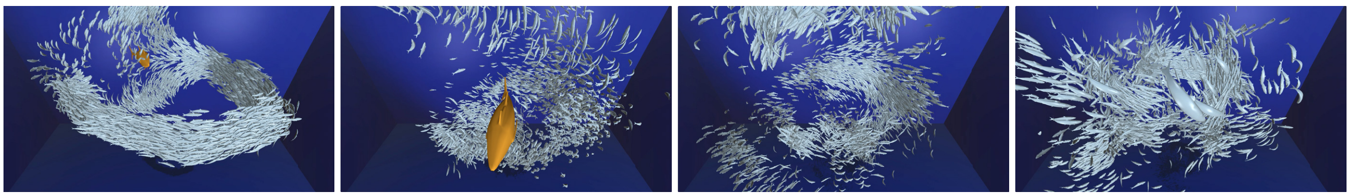

Figure 18: *A tuna plunges into a school of fish comprising 4,000 pilchatds swimming in a torus shape. The pilchatds conduct escape actions, and the school of fish momentarily disperses. However, once the tuna has left, the pilchards regroup to the original tube course.*

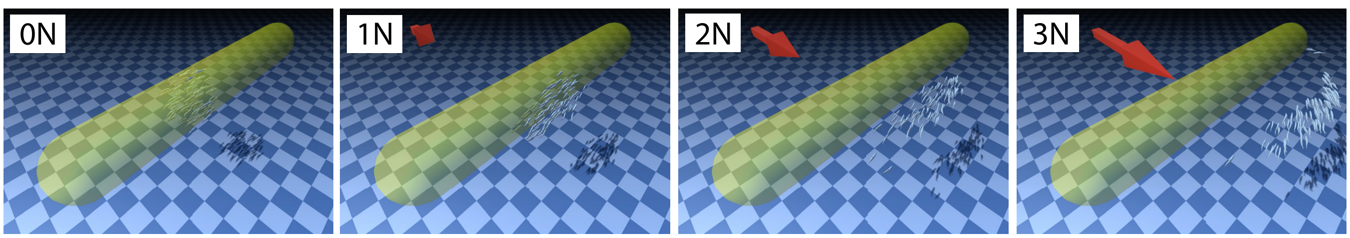

Figure 19: *Comparing the robustness with respect to external forces, such as water current. The red arrow represents the vector of the external force. As the external force increases, the virtual fish is not able to reach its target and is pushed outside the tube course.*

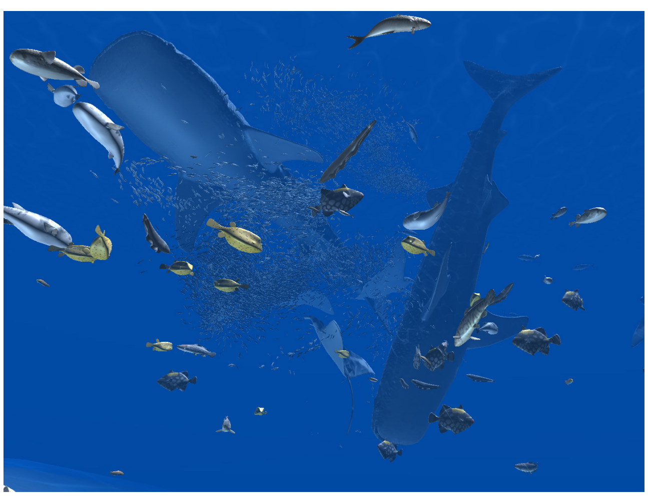

Figure 20: *A scene of a giant fish tank with 12,000 fish and 12 species of fish with texture mapping and better lighting.*

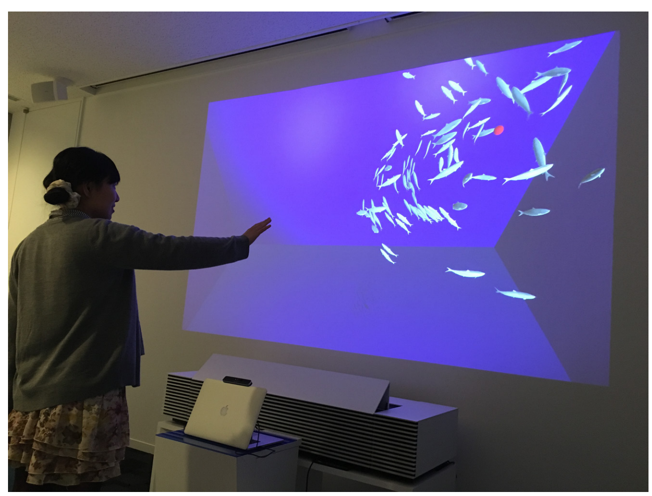

Figure 21: *A case example of the interactive application. A school of fish is controlled in real time by gestures using a 3D camera.*

behavior of actual fish.

Finally, in Figure 20, we show the simulated scene of a giant fish tank with 12 species of fish, including 12,000 pilchards, with texture mapping and better lighting.

7.3 Application

Since the proposed method works in real time, it can be implemented in interactive applications, such as games. As a simple example of an interactive application, we developed "Magic Aquarium," shown in Figure 21. In this application, an Intel RealSense

3D Camera is used to sense the position of the user's hands, and this is reflected in the positioning of the tube course for a school of fish projected onto the screen. Using this setup, the user can have the experience of maneuvering a school of fish.

8 Discussion

We showed that the swimming styles of various fish and the variations of the shapes of schools of fish can be recreated using a unified motion planner that instantly decides where to and how to swim. However, several limitations exist as well.

In the proposed method, the part of the fish that moves and the manner in which the part moves depends on the skeleton and skinning of the CG model. For example, the pectoral fins and dorsal fin of the Carangiform and Thunniform fish in this study and in reality will sometimes open and close with respect to the fish swimming movement. However, in this study, we judged that this motion has only a small effect on the appearance of the fish, deciding not to create these movements. To make the final animation more realistic by adding these factors, rigging is required for the pectoral fins and caudal fin with some customizing for the partial skeleton model and swimming form definitions. However, there are several useful aspects of the characteristics of this proposed method. By switching the swimming form definition depending on the distance of the camera, the level of detail control of the animation becomes possible, for example, having the smaller fins move when a fish is up close and only moving the body trunk when the same fish is distant.

In addition, our virtual fish do not have a physical output against the surrounding water environment. Due to this, it is difficult to recreate a scene where nearby aquatic plants or fins of other fish waver owing to the fish swimming. To recreate this type of fine movement in the proposed method, improvements are necessary, such as having the virtual fish output virtual force or using a partial combination with key frame animation.

9 Conclusion and Future Work

We propose the first animation-creation method that allows for a fish to instantly decide destination and speed. We considered a common mechanism for all fish when instantly making decisions about where to and how to swim. We used a unified motion planner that models this common mechanism and succeeded in recreating variations in swimming owing to skeletal differences or changes in conditions that can be observed in actual underwater scenes. It is easy to integrate our method into existing graphics pipelines. In addition, in the proposed method, it is possible to easily change the characteristics of movement by adjusting the parameters. The method also has a feature where the expression of the school of fish as a whole, such as tornado or circling, can be designated top-down.

We chose to simulate fish species based on the 12 types of swimming modes from Lindsey [Lindsey 1978]; however, there is a remarkable amount of variation in the appearance of fish, particularly in the construction of fins. For example, fish such as angelfish, goldfish, betta, and congo tetra have very large fins that waver beautifully like cloth when the fish swim. At present, with the proposed method, it is difficult to add these types of fin complexities to the animation; however, by adding a method where rigging and skinning is automatically conducted with respect to the fins, it may be possible to show fish realistically in real time as actual fish.

In this paper we simulated the schooling, avoidance and escaping behavior by behavior routines of the virtual fish. However, actual fish interact with the marine environment and can conduct more various behaviors, e.g., bottom-feeding flounder, responding to attacks, and predatory actions toward members of the same species based on territory. We can simulate these behaviors by expanding the behavior routines and constraint of the target probability distribution in our method. Moreover, we think that by modeling the elements of the marine environment that may affect the fish behavior, such as terrain, ocean current, and changes in water temperature, we should be able to simulate specific ocean areas, such as coral reefs.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. We thank Kenichiro Akimoto, Naoya Amata (STUDIO 4°C Co., Ltd.), and Masato Hirabayashi for their help on developing the proposed method. We also thank Masazumi Nakamura, Masayuki Akiba, Yoko Kubotera, Erika Tsuzuki, and Yoshihiko Ota (Intel Corporation) for their help on the demonstration of interactive application in this work. This work was funded by JSPS KAKENHI 15K12178 and HAYAO NAKAYAMA Foundation for Science & Technology and Culture H26-A1-98.

References

ARCHER, S., AND JOHNSTON, I. 1989. Kinematics of labriform and subcarangiform swimming in the antarctic fish. *Journal of Experimental Biology 210*, 143, 195–210.

BENTLEY, J. L. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM 18*, 9, 509–517.

BONE, Q., KICENIUK, J., AND JONES, D. 1978. On the role of the different fibre types in fish myotomes at intermediate swimming speeds. *Fishery Bulletin 76*, 691–699.

BOX, G. E. P., AND MULLER, M. E. 1958. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics 29*, 2, 610–611.

DOMENICI, P., AND BLAKE, R. W. 1997. The kinematics and performance of fish fast-start swimming. *The Journal of Experimental Biology 200*, 1165–1178.

FRIEDLANDER, A. M., AND PARRISH, J. D. 1997. Temporal dynamics of fish communities on an exposed shoreline in Hawaii. *Environmental Biology of Fishes 53*, 1, 1–18.

FUNGE, J., TU, X., AND TERZOPOULOS, D. 1999. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, 29–38.

GAESSER, G. A., AND BROOKS, G. A. 1984. Metabolic bases of excess post-exercise oxygen consumption: a review. *Medicine and Science in Sports and Exercise 16*, 1, 29–43.

GEIJTENBEEK, T., AND PRONOST, N. 2012. Interactive character animation using simulated physics: a state-of-the-art review. *Computer Graphics Forum 31*, 8, 2492–2515.

GRZESZCZUK, R., AND TERZOPOULOS, D. 1995. Automated learning of muscle-actuated locomotion through control abstraction. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, 63–70.

HOVE, J. R., O'BRYAN, L. M., GORDON, M. S., WEBB, P. W., AND WEIHS, D. 2001. Boxfishes (Teleostei: Ostraciidae) as a model system for fishes swimming with many fins: kinematics. *Journal of Experimental Biology 204*, 1459–1471.

HUDSON, R. C. 1973. On the function of the white muscles in teleosts at intermediate swimming speeds. *Journal of Experimental Biology 58*, 509–522.

JU, E., WON, J., LEE, J., CHOI, B., NOH, J., AND CHOI, M. G. 2013. Data-driven control of flapping flight. *ACM Transactions on Graphics 32*, 212.

KALLMANN, M., AND KAPADIA, M. 2014. Navigation meshes and real-time dynamic planning for virtual worlds. In *ACM SIGGRAPH 2014 Courses*.

KRUPCZYNSKI, P., AND SCHUSTER, S. 2008. Report fruit-catching fish tune their fast starts to compensate for drift. *Current Biology 18*, 24, 1961–1965.

LIGHTHILL, M. 1971. Large-amplitude elongated-body theory of fish locomotion. *Proceedings of the Royal Society B: Biological Sciences 179*, 1055, 125–138.

LINDSEY, C. 1978. Form, function, and locomotory habits in fish. In *Fish Physiology*, W. Hoar and D. Randall, Eds. Academic Press, New York, NY, United States, ch. 1, 1–100.

MASEHIAN, E., AND SEDIGHIZADEH, D. 2007. Classic and heuristic aproaches in robot motion planning – a chronological review. In *World Academy of Science, Engineering and Technology*, vol. 1, 101–106.

MASUDA, R. 2008. Seasonal and interannual variation of subtidal fish assemblages in Wakasa Bay with reference to the warming trend in the Sea of Japan. *Environmental Biology of Fishes 82*, 4, 387–399.

NELSON, J. S. 2006. *Fishes of the world*, 4th ed. Wiley, Hoboken, NJ, United States.

PENG, X. B., BERSETH, G., AND VAN DE PANNE, M. 2015. Dynamic terrain traversal skills using reinforcement learning. *ACM Transactions on Graphics 34*, 4.

RAYNER, M., AND KEENAN, M. 1967. Role of red and white muscles in the swimming of the skipjack tuna. *Nature 214*, 392–393.

REYNOLDS, C. W. 1987. Flocks, herds and schools: a distributed behavioral model. *Computer Graphics 21*, 4, 25–34.

RHODIN, H., TOMPKIN, J., KIM, K. I., AGUIAR, E. D., PFISTER, H., SEIDEL, H.-P., AND THEOBALT, C. 2015. Generalizing wave gestures from sparse examples for real-time character control. *ACM Transactions on Graphics 34*, 6.

SCHUSTER, S. 2012. Fast-starts in hunting fish: decision-making in small networks of identified neurons. *Current Opinion in Neurobiology 22*, 2, 279–284.

SI, W., LEE, S.-H., AND TERZOPOULOS, D. 2014. Realistic biomechanical simulation and control of human swimming. *ACM Transactions on Graphics 34*, 1.

SIMS, K. 1994. Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. 15–22.

STANTON, A., AND UNKRICH, L. 2003. Finding Nemo [motion picture]. *United States: Walt Disney Pictures*.

TAN, J., GU, Y., TURK, G., AND LIU, C. K. 2011. Articulated swimming creatures. *ACM Transactions on Graphics 30*, 4.

TAN, J., TURK, G., AND LIU, C. K. 2012. Soft body locomotion. *ACM Transactions on Graphics 31*, 4.

TERZOPOULOS, D., TU, X., AND GRZESZCZUK, R. 1994. Artificial fishes: autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life 1*, 4, 327–351.

TSUKAMOTO, K. 1984. Contribution of the red and white muscles to the power output required for swimming by the Yellowtail. *Bulletin of the Japanese Society of Scientific Fisheries 50*, 12, 2031–2042.

TSUKAMOTO, K. 1984. The role of the red and white muscles during swimming of the Yellowtail. *Bulletin of the Japanese Society of Scientific Fisheries 50*, 12, 2025–2030.

TU, X., AND TERZOPOULOS, D. 1994. Artificial fishes: physics, locomotion, perception, behavior. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*, ACM, New York, NY, United States, 43–50.

WALKER, J. A. 2000. Does a rigid body limit maneuverability? *Journal of Experimental Biology 203*, 22, 3391–3396.

WANG, H., HO, E. S. L., AND KOMURA, T. 2015. An energy-driven motion planning method for two distant postures. *IEEE Transactions on Visualization and Computer Graphics 21*, 1, 18–30.

WILLY, A., AND LOW, K. 2005. Development and initial experiment of modular undulating fin for untethered biorobotic AUVs. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 45–50.

YAMAGUCHI, Y. 2008. Aquanaut's Holiday: Hidden Memories [PlayStation 3]. *Japan: Sony Computer Entertainment*.