

Real-time Rendering of Aerodynamic Sound using Sound Textures based on Computational Fluid Dynamics

Yoshinori Dobashi[†]

Hokkaido University

Tsuyoshi Yamamoto[†]

[‡]The University of Tokyo

Tomoyuki Nishita[‡]

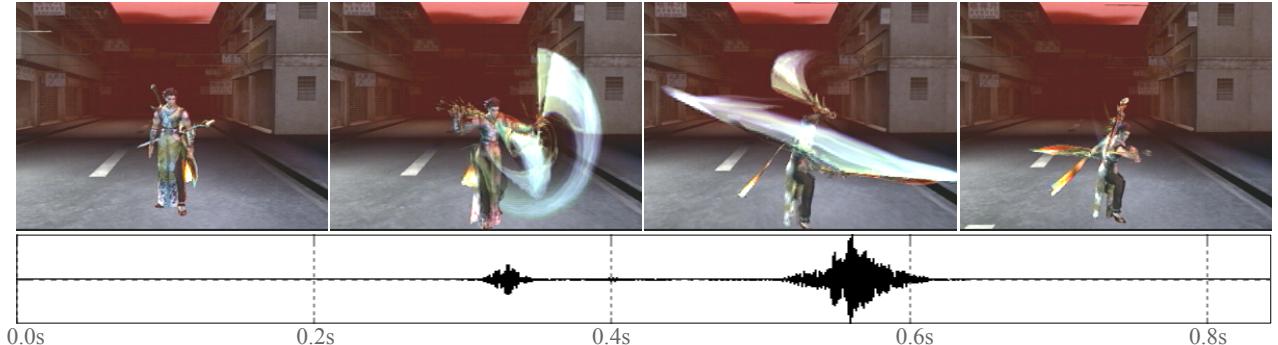


Figure 1: An example of aerodynamic sound generated by swinging swords. Sound wave (below) is computed based on the motion and shape of the swords. The motion blur is artificially added to visualize the motion of the swords.

Abstract

In computer graphics, most research focuses on creating images. However, there has been much recent work on the automatic generation of sound linked to objects in motion and the relative positions of receivers and sound sources. This paper proposes a new method for creating one type of sound called aerodynamic sound. Examples of aerodynamic sound include sound generated by swinging swords or by wind blowing. A major source of aerodynamic sound is vortices generated in fluids such as air. First, we propose a method for creating sound textures for aerodynamic sound by making use of computational fluid dynamics. Next, we propose a method using the sound textures for real-time rendering of aerodynamic sound according to the motion of objects or wind velocity.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Animation; I.6.8 [Simulation and Modeling]: Types of Simulation - Animation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.6.3 [Simulation and Modeling]: Applications; H.5.5 [Information Interfaces and Presentation]: Sound and Music Computing – Methodologies and techniques, Modeling.

Keywords: Sound Synthesis, Animation, Aerodynamic Sound, Simulation, Computational Fluid Dynamics.

[†]e-mail: {doba,yamamoto}@nis-ei.eng.hokudai.ac.jp
[‡]e-mail: nis@is.s.u-tokyo.ac.jp

Permission to make digital/hard copy of part of all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 2003 ACM 0730-0301/03/0700-0732 \$5.00

1. Introduction

One of the ultimate goals of computer graphics and virtual reality (VR) is realistic simulation of virtual environments. In addition to the images, sound is a very important element [Cook 2002], and adds information not included in the images. Therefore, many methods have been developed for the automatic generation of sound by making use of techniques developed in computer graphics, such as texture mapping/synthesis [Takala and Hahn 1992; Dubnov et al. 2002], beam tracing [Funkhouser et al. 1998], and rigid body simulations [O'Brien et al. 2002; van den Doel et al. 2001]. These methods make it possible to create realistic sound related to objects in motion and depending on the geometric relationship between the receiver and the sound source. Sound synthesis is developing into an important research field even in computer graphics.

Some of the more interesting sounds are the sounds generated by the motion of fluids such as wind and water. In computer graphics, although much research on the visualization of natural phenomena related to fluids such as smoke and water has been done, little attention has been paid to the sound of fluids. This paper focuses on one such sound - *aerodynamic sound*, with air as the fluid. In particular, we treat two types of aerodynamic sound, *aeolian tones* and *cavity tones*. The aeolian tones are usually generated when stick-like objects are present in a flow. Fig. 1 shows an example of aeolian tones generated by swinging swords. Cavity tones are generated by the flow around hollows (or cavities). The sound of a draft wind through a window is an example of cavity tones. The cause of aerodynamic sound is not the subtle oscillation of a solid object but vortices generated in the fluid (or air) [Fisher and Lowson 1971; Goldstein 1976]. Therefore, we need a new approach different than methods used for generating sound due to the oscillation of objects. One simple approach is to use recorded sound. However, the frequency and amplitude of aerodynamic sound depends on the shapes of the objects and the wind velocity. This makes it almost impossible to record aerodynamic sound for all cases.

In this paper, we propose a method for automatically generating aerodynamic sound. The advantages of our method are:

- (1) synthesis of sound corresponding to object motion and wind velocity,
- (2) synthesis of sound according to the object shape,
- (3) real-time rendering of sound,
- (4) consideration of stereo and Doppler effects.

In our method, pressure fluctuations on object surfaces are pre-computed using computational fluid dynamics (CFD) and are stored as sound textures. The sound textures are used for real-time rendering of the aerodynamic sound corresponding to the object motion or wind velocity. The user can specify the motion and wind velocity interactively.

In this paper, previously developed methods are discussed in Section 2. In Section 3, the principle and prediction of aerodynamic sound is explained. The basic idea of our method is described in Section 4. Then, in Sections 5 and 6 we propose methods for creating sound textures and real-time rendering of aerodynamic sound, respectively. Guidelines for setting the parameters for our method are described in Section 7, examples are demonstrated in Section 8, and the limitations of our method are discussed in Section 9. Finally, Section 10 gives our conclusions.

2. Related Work

In computer graphics, methods for generating sound based on physical simulation can be classified into two groups. One is the simulation of the propagation of sound, and the other is for the synthesis of sound.

In the first group, reflection and absorption of sound due to surrounding objects are simulated. This makes it possible to compute sound taking into account the geometric relationship between the receiver and the sound source [Takala and Hahn 1992; Funkhouser et al. 1998; Funkhouser et al. 1999; Tsingos et al. 2001; Lokki et al. 2002]. These methods, however, do not treat the synthesis of sound waves radiated from sound sources.

The methods in the second group compute sound waves generated by sound sources. Hahn et al. proposed a method for creating sounds procedurally by controlling sound parameters so that the sound changes as a result of changes in the object motion [Hahn et al. 1995]. A method using a sound map for synthesizing sound is proposed by [van den Doel and Pai 1998]. However, the purpose of that method is to simulate contact sounds, so it is not applicable to aerodynamic sound. O'Brien et al. developed methods for computing sound waves by numerical analysis of the subtle oscillation of objects taking into account their shape and material [O'Brien et al. 2001; O'Brien et al. 2002]. These methods can automatically generate the sound of objects colliding. They focus on the sound generated by the oscillation of solid objects. However, the sound of fluids including aerodynamic sound is not taken into account.

On the other hand, researchers in the field of CFD have developed methods for predicting aerodynamic sound [Tam 1995; Lele 1997]. The purpose of this research was to reduce the noise due to high-speed transportation facilities and air-conditioners, etc. In this field, in order to predict the sound precisely, the motion of the objects and the changes in wind velocity are taken into account in the numerical fluid analysis. However, this analysis is very complex and requires a large amount of computation. Therefore, this it is not appropriate for real-time applications such as VR.

The method proposed in this paper generates aerodynamic sound in real-time by making use of the predictive methods developed in the field of computational fluid dynamics.

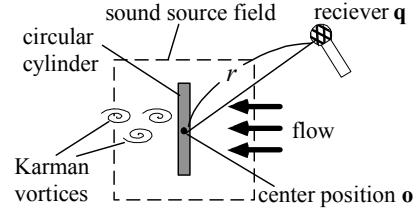


Figure 2: Generation of aeolian tones.

3. Curle's Model for Prediction of Aerodynamic Sound

The main source of aerodynamic sound is vortices in fluids. For example, the principle of generation of aeolian tones is as follows. As shown in Fig. 2, when a stick-like object is placed in a flow, vortices called Karman vortices are periodically generated behind the object. At the time the vortices are created, subtle fluctuations of pressure occur. These subtle fluctuations are perceived as sound. Therefore, the primary frequency of aerodynamic sound corresponds to the frequency of generation of vortices. In 1878, Strouhal performed an experiment with a circular cylinder and found the relationship, $fD/v = S_t$, where f is the frequency, v the speed of the flow, and D the diameter of the cylinder [Strouhal 1878]. S_t is called the Strouhal number and is about 0.2 for the circular cylinder.

A basic theory for aerodynamic sound was established by Lighthill in 1952 [Lighthill 1952]. Recent research on the numerical prediction of aerodynamic sound is based on this theory [Tam 1995; Lele 1997]. One of the most straightforward approaches is the numerical analysis of the subtle fluctuation of fluids (air in most cases) by solving the *compressible* Navier-Stokes equations. However, this analysis is computationally very expensive since the magnitude of the fluctuations is of the order of 10^{-5} compared to the atmospheric pressure and we need a dense mesh and very small time step to capture such subtle fluctuations.

To address this problem, our method makes use of a model developed by Curle [Curle 1953]. In this model, the sound pressure p_α at the receiver can be computed if the behavior of air around the sound source is known. When the receiver is sufficiently distant from the sound source in comparison with the length of the sound wave and the size of the object is sufficiently small relative to the length of the sound wave, the sound pressure p_α is expressed by the following equation.

$$p_\alpha(\mathbf{q}, t) = \frac{1}{4\pi c_0 r^2} (\mathbf{q} - \mathbf{o}) \cdot \mathbf{g}(t - r/c_0), \quad (1)$$

$$\mathbf{g}(t) = \left(\frac{\partial}{\partial t} \int_S n_x p(\mathbf{s}, t) dS, \frac{\partial}{\partial t} \int_S n_y p(\mathbf{s}, t) dS, \frac{\partial}{\partial t} \int_S n_z p(\mathbf{s}, t) dS \right) \quad (2)$$

where, as shown in Fig. 2, \mathbf{q} is the position of the receiver, \mathbf{o} the center position of the object (such as the center of gravity), t the time, c_0 the sound speed, p the air pressure on the object surface, r the distance between points \mathbf{q} and \mathbf{o} , \mathbf{s} the position on the object surface S , and (n_x, n_y, n_z) the normal vector at point \mathbf{s} . In Eq. 1, the term $t - r/c_0$ indicates the time delay of the sound due to the distance between the source and the receiver. The vector function \mathbf{g} in Eq. 2 is independent of the position of the receiver. In this paper, we call the function, \mathbf{g} , a *sound source function*. Although Curle's model gives just an approximate solution to the aerodynamic sound, several experiments indicate the validity of the model [Phillip 1956; Gerrad 1961; Roshko 1961; Morkovin 1964].

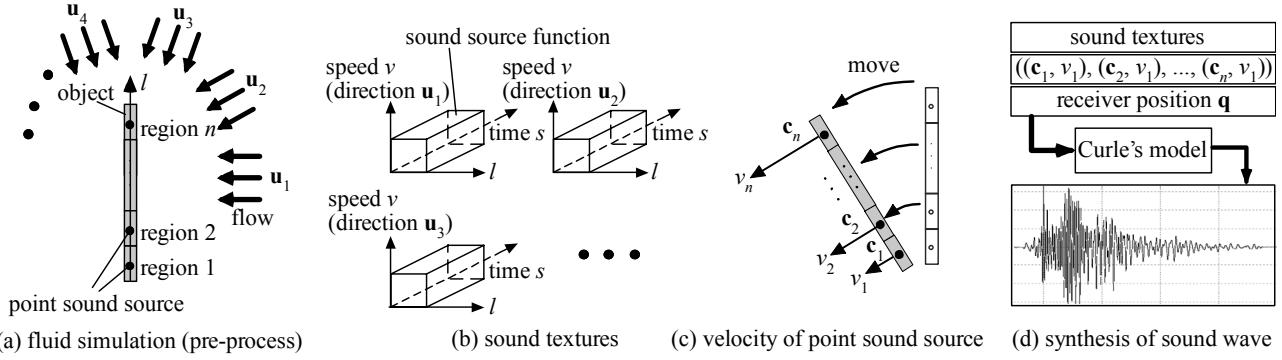


Figure 3: Synthesis of aerodynamic sound by our method.

By using Curle’s model, the sound pressure at the receiver is obtained if we can compute the pressure fluctuations on the surface of the object. So first, the pressure p on the object surface is computed by numerical analysis of the *incompressible* Navier-Stokes equations. Next, the sound source function is evaluated by Eq. 2. Then, the sound pressure p_α is obtained from Eq. 1.

4. Basic Idea of Our Method

It is very difficult to apply Curle’s model in a straightforward way to real-time simulation of an aerodynamic sound generated by a swinging sword (aeolian tone) or a draft wind (cavity tone). This is because precise numerical analysis of a fluid taking into account the motion of an object or the change in wind velocity is required, and this requires a large amount of computation. To address this problem, we first compute the behavior of flow through a static object. Using the results of fluid analysis, the sound source function \mathbf{g} is computed and stored as sound textures. The sound textures are used to compute the aerodynamic sound corresponding to the motion of the object or wind velocity. For aeolian tones, we assume the sounds are not occluded by the object itself, such as stick-like objects and spheres. For cavity tones, we assume the cross sectional shapes of the hollows (or cavities) are almost uniform. In the following, we explain our method for the generation of aeolian tones. The same idea can be applied to cavity tones.

To apply Curle’s model, the size of the object must be sufficiently small relative to the length of the sound wave. Therefore, the object surface is divided into n regions as shown in Fig. 3. The aerodynamic sound is computed approximately by applying Curle’s model to each of the regions and by summing the resulting sound. This is equivalent to assuming that there are n independent virtual point sound sources at the centers of the regions (Fig. 3(a)). Let us denote the sound source function of the l th region as \mathbf{g}_l . The position number of each point sound source is denoted by l as shown in Fig. 3(a). The number of regions (or point sound sources) n is specified by the user. See Section 7 for the appropriate number of regions.

The sound source functions \mathbf{g}_l are computed in a pre-process and stored as sound textures. The aerodynamic sound is different depending on the direction in which the object is swung in or the direction of the wind velocity. Therefore, we analyze the flow fields for various sample directions of the flow \mathbf{u} and speed v using CFD (Fig. 3(a)). Then, the sound source function \mathbf{g}_l corresponding to each point sound source is stored. See Section 7 about choosing the appropriate speed v and the number of sampling directions.

The aerodynamic sound in response to an object in motion is computed in real-time by using the sound textures. First, as shown

sound texture: $\mathbf{w}(l, s, \mathbf{u}, v)$
a table that stores the sound source function of a region corresponding to point sound source l at time s . Each element of the table stores three components of the sound source function. $0 < l < L$, $0 < s < T$, $0 < v < V$, where L , T , V are the sizes of the texture.

Figure 4: Definition of sound texture.

in Fig. 3(c), the direction \mathbf{c}_l and speed v_l of the velocity vector of each point sound source is computed based on the motion of the object. Next, the values of the sound source functions corresponding to \mathbf{c}_l and v_l are computed using the sound textures. Then, the sound pressure due to each point sound source is computed using Curle’s model (Eqs 1. and 2.). The final sound pressure at the receiver is obtained by summing these pressures. This process is repeated at short time intervals to create aerodynamic sound waves (see Fig. 3(d)).

In the following, the sound texture is expressed as $\mathbf{w}(l, s, \mathbf{u}, v)$. The definition of sound texture is shown in Fig. 4. In this paper, to avoid confusion, s indicates the time in the texture domain (or texture coordinate) and t indicates the actual time.

5. Sound Texture of Aerodynamic Sound

This section describes the method for computing sound textures. Three properties of aerodynamic sound are used to reduce the computational cost for the numerical fluid analysis.

5.1 Efficient Computation using the Properties of Aerodynamic Sound

To compute sound textures, numerical fluid analysis must be repeated for many speeds and sampling directions (see Figs. 3(a) and (b)). This results in a very long computation time. To reduce the time, we make use of the following properties of aerodynamic sound [Fisher and Lawson 1971; Goldstein 1976; Phillips 1956; Rockwell 1977].

- 1) The frequency of the sound is proportional to the flow speed v .
- 2) The amplitude of the sound is proportional to the 6th power of the flow speed v . For cavity tones, the amplitude is proportional to the 4th power of v .

Using these properties, we can compute the sound source function of an arbitrary flow speed if the sound texture at the base speed v_0 , $\mathbf{w}(l, s, \mathbf{u}, v_0)$, has been obtained. Moreover, for stick-like objects as shown in Fig. 5, the following property is utilized [Williamson 1988; Lee and Budwig 1991].

- 3) In the case of a stick-like object inclined at an angle ϕ to the direction of the flow, the sound is roughly equivalent to that of an object perpendicular to a flow of speed $v \cos \phi$ (see Fig. 5)¹.

Therefore, for sampling directions of the flow \mathbf{u} we need only consider the directions perpendicular to the axis \mathbf{n} of stick-like objects (see Fig. 5). These properties can reduce both the computational cost and the memory requirements drastically. Note that the sound source functions computed by using the above properties are not always correct because of the turbulence of the flow. The characteristics of fluid dynamics depend on the Reynolds numbers. As the Reynolds number becomes larger, the flow becomes more turbulent. Our method cannot take this turbulence into account. However, this turbulence is perceived as noise that is not favorable for sound effects. As shown in Section 8, we can obtain satisfactory results by using the above method.

5.2 Numerical Fluid Simulation

We use either a finite difference method or a finite element method for the numerical fluid simulation. The finite element method is more accurate than the finite difference method. However, satisfactory results can be obtained even by the finite difference method. The finite difference method we use is well described in [Foster and Metaxas 1997; Stam 1999]. For the finite element analysis, we use a commercial program, STAR CD developed by Computational Dynamics Ltd. The analysis method is Streamline-Upwind/Petrov-Galerkin method [Brooks and Hughes 1982]. This is one of the common methods for fluid analysis. For more details, please refer to technical books such as [Chung 2002]. The physical parameters required for the fluid analysis are the viscosity coefficient and the density of the atmosphere. When the temperature is 20 degrees centigrade and the atmospheric pressure is 760 [mmHg], the density and the viscosity are 1.025 [kg/m³] and 18.0 × 10⁻⁶ [Pa·s], respectively. The time step for the simulation should be sufficiently short to capture the subtle fluctuations. It depends on individual situations but our recommendation is 1/8000 [s] or 1/16000 [s].

The numerical simulation is repeated for each of m sampling directions of the flow, \mathbf{u}_j ($j = 1, \dots, m$). We rotate the object instead of changing the flow direction. At each time step in the simulation, the sound source function is computed using Eq. 2 and stored as the sound texture. The numerical simulation is computed for a specified number of time steps. During the early steps, however, the simulation is numerically unstable. Therefore, computation of the sound source function begins after the simulation has reached a stable state.

The following subsections describe computational conditions for the generation of the sound textures.

5.2.1 Choosing 2D or 3D simulation

We choose 2D or 3D simulation depending on the shape of the object. When the cross sectional shape of the object is almost uniform such as a circular cylinder as shown in Fig. 6(a), we choose 2D simulation based on the cross sectional shape. This decreases the computation time. In this case, a sound texture with $l = 0$ is obtained, $\mathbf{w}(0, s, \mathbf{u}, v_0)$. This sound texture is applied to all the point sound sources to compute the aerodynamic sound. Experimentally, we heard no significant difference between 2D and 3D simulations for this kind of object. On the other hand, it is

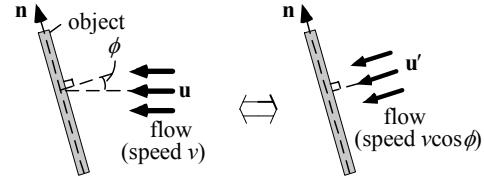


Figure 5: Property for inclined object.

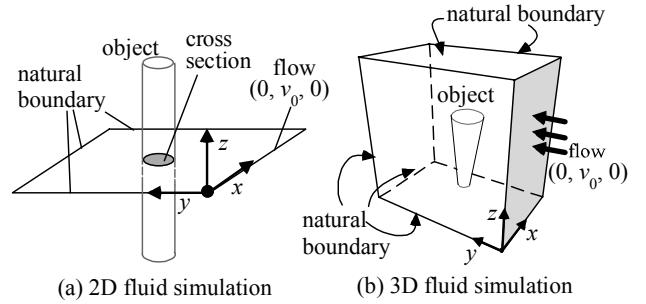


Figure 6: Conditions for fluid simulation.

difficult to apply 2D simulation to an object whose cross sectional shape is not uniform, as shown in Fig. 6(b). In this case, 3D simulation is used to compute the sound texture, $\mathbf{w}(l, s, \mathbf{u}, v_0)$. Note that the base speed v_0 as well as the sound texture is stored.

5.2.2 Boundary Conditions

The boundary conditions in the fluid simulation are as follows.

First, in 2D simulations, the cross sectional shape is computed (Fig. 6(a)). Then the behavior of the fluid is computed in the xy plane to compute the velocity $\mathbf{v} = (v_x, v_y, v_z)$ and the pressure p , where v_z is always 0 in 2D simulations. For one of the boundary conditions, the velocity at $y = 0$ is set to $(0, v_0, 0)$. A natural boundary condition is applied to the rest of the boundaries. That is, the differentials of velocity and pressure normal to the boundaries are 0. At points on the surfaces of the object, the velocities are 0 and the differentials of pressure in the normal directions are 0. Initially, all the velocities and pressures in the region being analyzed are set to 0.

For 3D simulations, the initial status and boundary conditions are similar to those for 2D simulations (see Fig. 6(b)).

5.3 Memory Requirements

This subsection discusses the memory requirement for the sound textures. The sound texture is a function of (l, s, \mathbf{u}, v) . Since we use the three properties described in Section 5.1, the speed of the flow v is fixed at v_0 (base speed). This means the sound texture is represented by a three-dimensional table. However, if we use 2D fluid simulation, l is fixed (see Section 5.2.1). In this case, the sound texture is represented by a two-dimensional table. Besides, if the shape of the object is symmetric such as circular cylinders, \mathbf{u} can also be fixed. Then, one-dimensional table is sufficient. In this case, the memory requirement is less than tens of kilobytes. For other cases, the memory requirement is proportional to the number of the sampling directions of the uniform flow. If we use 3D fluid simulation, it is proportional to the number of the point sound sources, too.

6. Real-time Rendering of Aerodynamic Sound

This section describes the method for computing sound waves using sound textures. The basic idea is similar to the one called

¹ Strictly speaking, this property is true only when $\phi < 35$ degrees and the Reynolds number is less than around 10⁴.

wavetable synthesis [van den Doel et al. 2001]. That is, a precomputed wavetable (or sound texture in our case) is resampled for real-time synthesis of sound. However, this approach is not directly applicable to our case since we have to take into account the properties of the aerodynamic sound in resampling the table. The detail of our method is as follows.

Let us assume the object is static at time $t = 0$ when the process starts. The value of the sound source function is zero at $t = 0$. Computation of the sound pressure is repeated at short intervals of time, Δt . In the following, $t_k = k\Delta t$, where k is a non-negative integer indicating time steps in the simulation.

Let us consider a case where time $t = t_k$. First, the velocity vectors of each point sound source are computed based on the motion of the object. Let us denote the direction and speed of the velocity vector of a point sound source l as $\mathbf{c}_l(t_k)$ and $v_l(t_k)$, respectively. Next, the sound source functions are computed using the sound textures. Using the properties 1) and 2) described in Section 5.1, the sound source function is computed from the following recurrence relation.

$$\begin{cases} s_k = (v_l(t_k)/v_0\Delta t) + s_{k-1} \\ \mathbf{g}_l(t_k) = (v_l(t_k)/v_0)^6 \mathbf{w}(l, s_k, \mathbf{c}_l, v_0) \end{cases}, \quad (3)$$

where $s_0 = 0$. As shown in Eq. 3, the texture coordinate s_k at time t_k is recursively computed from s_{k-1} , the value stored in the sound texture \mathbf{w} at texture coordinate s_k is used to obtain $\mathbf{g}_l(t_k)$. When the sound textures are computed by 2D simulations for stick-like objects, the property 3) in Section 5.1 is utilized as follows. That is, $\mathbf{c}'_l(t_k)$ and $v_l(t_k)\cos\phi_l$ are used instead of $\mathbf{c}_l(t_k)$ and $v_l(t_k)$ in Eq. 3, where $\mathbf{c}'_l(t_k)$ is obtained by projecting $\mathbf{c}_l(t_k)$ onto a plane whose normal \mathbf{n} is in the direction of the object axis and normalizing to a unit vector, and ϕ_l is the angle between the vectors \mathbf{n} and $\mathbf{c}_l(t_k)$ (see Fig. 5).

As shown in Fig. 7, Eq. 3 indicates that the sound source function sampled at intervals of $(v_l(t_k)/v_0)\Delta t$ in the texture domain is mapped onto the actual time domain. This implies that the frequency of the sound becomes $(v_l(t_k)/v_0)$ times higher (or lower). Clearly, the amplitude becomes $(v_l(t_k)/v_0)^6$ times larger. The sound texture is used periodically since the texture size T is finite (see Fig. 7). The ends of the texture are blended to give a smooth transition of the sound source function.

The final sound pressure at the receiver is computed by summing the sound pressure due to each point sound source, that is,

$$p_v(\mathbf{q}, t_k) = \sum_{l=1}^n \frac{1}{4\pi c_0 r_l^2} (\mathbf{q} - \mathbf{o}_l) \cdot \mathbf{g}_l(t_k - r_l/c_0), \quad (4)$$

where r_l is the distance from the receiver to the sound source l , and \mathbf{o}_l is the position of the sound source l . Before the sound reaches the receiver, $t_k - r_l/c_0 < 0$. In this case, we assume $\mathbf{g}_l(t_k - r_l/c_0) = \mathbf{0}$. The sequence of sound pressures is the resulting sound wave.

We use a sound library developed by Banno [Banno 2002] to control the audio devices. Images are rendered by using OpenGL. Stereo sounds are easily created by placing two receivers at the positions of ears. In this case, sound pressures are computed twice, once for each ear.

7. Notes on Setting Parameters

Apart from the parameters for fluid simulation, there are three parameters that have to be specified by the user in our method. These are 1) the number of point sound sources (n), 2) the base speed (v_0), and 3) the number of sampling directions of flow (m) for creating sound textures. This section gives some guidelines for

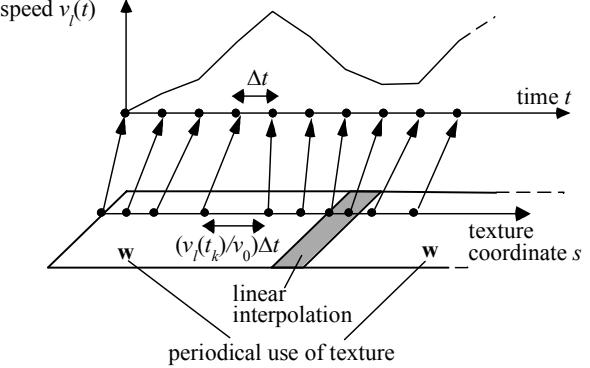


Figure 7: Mapping of sound textures.

setting these parameters. We omit the parameters for fluid simulation. For this, please refer to technical books in the field of CFD, for example, [Chung 2002].

7.1 Number of Point Sound Sources

An inappropriate number of point sound sources would result in unnatural aerodynamic sound. The number is determined by the interval between point sound sources and there are appropriate intervals. We can assume that pressure fluctuations in a range $[-\Delta h/2, +\Delta h/2]$ around a point sound source are roughly the same since a fluid is a continuous medium. The length Δh is called a correlation length [Fisher and Lowson 1971; Goldstein 1976]. We use the correlation length for the interval between point sound sources. The correlation length depends on the Reynolds number. In our application, such as aerodynamic sound generated by swinging swords or wind blowing, the Reynolds number ranges approximately from 10^4 to 10^5 . In this case, the correlation length is around $3D$, where D is a dimension of the object [Phillips 1956; Gerrard 1961; Fisher and Lowson 1971; Goldstein 1976]. For example, in the case of a swinging circular cylinder, we can use the diameter of its cross-section as D . For reference, Reynolds number Re is computed from the following equation.

$$Re = DU/\eta, \quad (5)$$

where U is the average speed of the flow and η is the dynamic viscosity. In the case of air, η is roughly 1.5×10^{-5} [m²/s].

7.2 Base Speed

If the base speed v_0 is too large, the Reynolds number becomes large and the flow becomes turbulent. This produces unpleasant noises in the resulting sound. We recommend choosing the base speed so that Reynolds number is from 10^3 to 10^4 . In our experiment, when using such a base speed, the noise due to turbulence is not perceived. Actually, we used $v_0 = 10$ [m/s] for all the examples shown in Section 8.

7.3 Number of Sampling Directions

A single sampling direction ($m = 1$) is sufficient when the shape of the object is isotropic such as for a circular cylinder or sphere. However, in the case of objects like swords, multiple directions have to be sampled at sufficiently small intervals. If the number of sampling directions is too few, the aerodynamic sound is perceived as a chord of sounds corresponding to neighboring sampling directions. In our current implementation, we choose 5 degrees as the sampling interval. A possible approach for automatic computation of the number is that m is incremented until the sound source functions for neighboring sampling directions become almost the same.

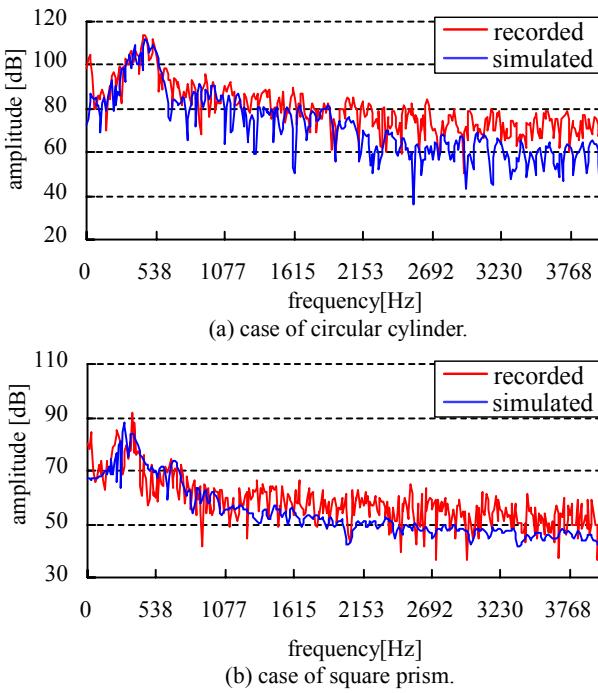


Figure 8: Comparison of real sound with simulated sound.

8. Results

First, simulated aerodynamic sound is compared with the real (recorded) sound. Then, several examples are demonstrated.

8.1 Experimental Results

To verify the validity of our method, we recorded the aerodynamic sound from swinging a circular cylinder and a square prism. The diameter of the circle and the side length of the square were 1 cm. The lengths of the cylinder and prism were 50 cm. While we recorded the sound, the motion of the objects was measured by a 3D motion tracker. We compared the real sound with the sound simulated using the measured motion. In the simulation, the sound textures were generated by 2D fluid simulations. We used the finite element method. The precomputation time was 45 minutes for the circular cylinder. For the square prism, the number of sampling directions was 40 in total. However, we only needed to consider one eighth of them (i.e. 8 directions) due to the symmetrical shape of the prism and it took 4 hours. We used a SGI Origin 2000 with 32 processors (4 processors are used for our simulation). The sizes of the sound textures are 40 KB (circular cylinder) and 200 KB (square prism) approximately. The number of sound sources was 16 since the correlation length in this case was $3.0D = 3$ cm.

Fig. 8 shows spectra of the sounds. Fig. 8(a) corresponds to the circular cylinder and Fig. 8(b) to the square prism. The red lines correspond to the recorded sounds and the blue lines to simulated sounds. As shown in Fig. 8, the amplitude of the high frequency components of the real sound is larger than that of the simulated sound. The reason for this is that our method is unable to take turbulence into account when the Reynolds number is large. However, the spectral distributions around the peak frequencies (462 Hz in the circular cylinder and 344 Hz in the square prism) are almost the same. This implies that our method captures the distinctive features of the aerodynamic sound.

8.2 Examples of Aerodynamic Sound

Fig. 9 shows an example of the aerodynamic sound generated by swinging a sword. The sound textures are generated by 2D fluid simulations. Fig. 9(a) shows the cross sectional shape of the sword. The length of the sword is 70 cm. The number of point sound sources is 10. The motion of the sword is specified by using a 3D motion tracker. The user swings the tracker instead of the sword to input the motion into our system. Fig. 9(b) shows the motion of the sword. Fig. 9(c) shows the speed of the end of the sword and the corresponding sound wave. The amplitudes of the sound waves are normalized so that the maxima are 1.0. Fig. 9(d) shows the spectrum of the sound. As shown in Fig. 9(c), the sound changes according to the motion of the sword.

In Fig. 10, we use a club instead of the sword. The shape of the club is shown in Fig. 10(a). In this case, 3D fluid simulations are used to create the sound textures. The number of point sound sources is 10. In contrast to Fig. 9, low frequency components are dominant. This indicates that the aerodynamic sound is different depending on the shapes of the object.

Next, Fig. 11 shows a simulation of the aerodynamic sound generated by rotating a sphere tied with a wire. Fig. 11(b) shows the sound waves when the sphere is rotated four times. The sphere is rotated at a constant speed. The sound textures are generated by 3D fluid simulation. The aerodynamic sound generated by the wire is not taken into account. Doppler effects due to the motion of the sphere are simulated. This makes the frequency higher when the sphere approaches the receiver and vice versa.

Fig. 12 shows the aerodynamic sound generated by throwing a circular cylinder at the receiver. The cylinder rotates as it approaches. Doppler effects are taken into account.

Next, the proposed method is applied to character animations. In Fig. 1, a warrior moves with his swords swinging. In this example, we artificially add echoes due to the surrounding walls by filtering the obtained sound waves. We can create compelling animations by adding aerodynamic sound corresponding to the motion of the warrior. Fig. 13 shows another example where a bear swings a huge club. In these examples, we generate aerodynamic sound by scaling the speed. In this way, it is possible to emphasize aerodynamic sound using our method.

Finally, Figs. 14 and 15 show examples of aerodynamic sound generated by wind blowing. Fig. 14 shows an example of sound due to a cold wintry wind through a fence. The fence is modeled by circular cylinders. Fig. 15 shows an example of cavity tones generated by a draft through a gap between windows. In these examples, snow is rendered in order to visualize the wind velocity and has no effects on the sound. The user specifies the velocity of the wind interactively at each time step. Then small random numbers are added to the wind velocity. The aerodynamic sound is generated by using the wind velocity as the velocity of each point sound source. Reasonable sounds are generated corresponding to the wind velocity.

In the examples shown in this subsection, the shapes of objects were created by using our interactive modeling tool. The images were rendered in real-time, 60 frames per second. The corresponding sounds were generated simultaneously. The computation times for the sound textures took about one hour (2D simulations) and 10 hours (3D simulations). We used the finite difference method. The sizes of the sound textures ranged from 40 KB to 5 MB. We used a desktop PC with a Pentium III (1.3 GHz) processor. Please refer to the conference DVD (or our website) for movies corresponding to the examples shown in this section.

9. Discussion

Users of our method should pay attention to the following three points.

First, as mentioned previously, our method cannot take into account randomness due to turbulence of the flow. That is, our method cannot reproduce high frequency components due to turbulence. However, our method does reproduce frequency components around the peak frequency. So, it captures the distinctive shape-dependant features and is useful for sound effects. The high frequency components may be added by using noise models that are often used for sound synthesis of wind instruments [Cook 2002]. If high frequency components must be reproduced exactly, the properties described in Section 5.1 should not be used. Instead, a number of sound textures corresponding to various velocities should be generated. Moreover, 3D fluid simulation is also recommended since there is turbulence that 2D simulation cannot capture. These enhancements would improve the results. However, they will increase the computation time significantly.

Secondly, vortices generated by the end of the stick-like object have an effect on the aerodynamic sound. This effect is not taken into account when 2D fluid simulations are used to generate sound textures. 3D fluid simulations are required to take into account this effect even if the object shapes are stick-like.

Thirdly, the sound is occluded by the object itself if the object occludes itself, such as with hammers. Curle's model cannot take this into account. Although we can compute reasonable aerodynamic sound even in this case, the result would not be completely correct.

As described in the above discussion, our method is an approximate solution to the actual physical phenomena. However, our method is sufficient for sound effects as demonstrated by the examples shown in Section 8.

Finally, let us discuss human interventions involved in our method. The sound textures are automatically computed after specifying shape of object, positions of the point sound sources, the sampling directions of the uniform flow, and parameters for the fluid simulation. Once the sound textures have been generated, the user specifies the motion of the object interactively by using input devices such as a mouse and a motion tracker. The aerodynamic sound is automatically generated according to the specified motion.

10. Conclusion

In this paper, we have proposed a new method for generating aerodynamic sound in real-time.

In the method, sound textures for aerodynamic sound are created. The sound textures store the sound source function, that is, the pressure fluctuations at the surface of the object. They are computed by making use of computational fluid dynamics, taking into account the shape of object. We utilized the properties of aerodynamic sound to drastically reduce the computation time and memory requirements for the sound textures.

Furthermore, we have realized real-time rendering of sound using sound textures. The sound source function is evaluated according to the object motion or wind velocity.

In conclusion, our method adds a new element, aerodynamic sound, to improve the realistic simulation of virtual environments.

One of the most interesting tasks for the future is to develop a method for synthesizing the sound in other fluids such as water.

Acknowledgements

The authors would like to acknowledge Atsushi Kunimatsu (Broadband System LSI Project, TOSHIBA Corp., Japan), Tsunemi Takahashi (Corporate R&D Center, Toshiba Corp., Japan), and Naofumi Shibata (TCI Division, Toshiba Information Systems Corp., Japan) for their support in the fluid analysis, and Kazunori Yamamoto and Osamu Inoue (TAITO Corp., Japan) for their help in producing the examples of the character animations. We would also like to thank Prof. Nelson Max (University of California) for his helpful comments.

References

- BANNO, H. 2002, <http://www.itakura.nuee.nagoya-u.ac.jp/people/banno/index.html>
- BROOKS, A. AND HUGHES, T. J. R. 1982, Streamline-Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, *Computer Methods in Applied Mechanics and Engineering*, 32, 199-259.
- COOK, P. R. 2002, Sound Production and Modeling, *IEEE Computer Graphics & Applications*, 22, 4, 23-27.
- COOK, P. R. 2002, Real Sound Synthesis for Interactive Applications, A K Peters.
- CURLE, N. 1953, The Influence of Solid Boundaries Upon Aerodynamic Sound, In *Proceedings of Royal Society London*, A211, 569-587.
- CHUNG, T. J. 2002, Computational Fluid Dynamics, Cambridge University Press.
- DUBNOV, S., BAR-JOSEPH, Z., RAN, E., LISCHINSKI, D., AND WERMAN, M. 2002, Synthesizing Sound Textures through Wavelet Tree Learning, *IEEE Computer Graphics & Applications*, 22, 4, 38-48.
- FISHER, M. J. AND LOWSON, M. V. 1971, Aerodynamic noise, *Journal of Fluid Mechanics*, 48 (Part 3), 593-603.
- FUNKHOUSER, T., CARLBOM, I., ELKO, E., PINGALI, G., SONDHI, M., AND WEST, J. 1998, A Beam Tracing Approach to Acoustic Modeling for Interactive Virtual Environments, In *Proceedings of ACM SIGGRAPH 98*, Annual Conference Series, 21-32.
- FUNKHOUSER, T., MIN, P., AND CARLBOM, I. 1999, Real-time Acoustic Modeling for Distributed Virtual Environment, In *Proceedings of ACM SIGGRAPH 99*, Annual Conference Series, 365-374.
- FOSTER, N. AND METAXAS, D. 1997, Modeling the Motion of a Hot, Turbulent Gas, In *Proceedings of ACM SIGGRAPH 97*, Annual Conference Series, 181-188.
- GERRAD, J. H. 1961, An Experimental Investigation of the Oscillating Lift and Drag of a Circular Cylinder Shedding Turbulent Vortices, *Journal of Fluid Mechanics*, 2, 244-256.
- GOLDSTEIN, M. E. 1976, Aeroacoustics, McGraw-Hill.
- HAHN, J., GEIGL, J., LEE, J., GRITZ, L., TAKALA, T., AND MISHRA, S. 1995, An Integrated Approach to Sound and Motion, *Journal of Visualization and Computer Animation*, 6, 2, 109-123.
- LEE, T. AND BUDWIG, R. 1991, A Study of the Effect of Aspect Ratio on Vortex Shedding behind Circular Cylinders, *Physics of Fluids*, A3, 2, 309-315.
- LELE, S. K. 1997, Computational Aeroacoustics: A Review, *American Institute of Aeronautics and Astronautics Paper (AIAA Paper)*, 97-0018.

- LIGHTHILL, M. J. 1952, On Sound Generated Aerodynamically: I. General Theory, In *Proceedings of Royal Society London*, A221, 564-587.
- LOKKI, T., SAVIOJA, L., VAANANEN, R., HUOPANIEMI, J., AND TAKALA, T. 2002, Creating Interactive Virtual Auditory Environments, *IEEE Computer Graphics & Applications*, 22, 4, 49-57.
- MORKOVIN, M. V. 1964, Flow around a Circular Cylinder – a Kaleidoscope of Challenging Fluid Phenomena, In *Proceedings of ASME Symposium on Fully Separated Flows*, 102-118.
- O'BRIEN, J. F., COOK, P. R., AND ESSL, G. 2001, Synthesizing Sounds from Physically Based Motion, In *Proceedings of ACM SIGGRAPH 2001*, Annual Conference Series, 529-536.
- O'BRIEN, J. F., SHEN, C. AND GATCHALIAN, C. M. 2002, Synthesizing Sounds from Rigid-Body Simulations, In *Proceedings of ACM SIGGRAPH 2002 Symposium on Computer Animation*, 175-181.
- PHILLIPS, O. M. 1956, The Intensity of Aeolian Tones, *Journal of Fluid Mechanics*, 1, 607-624.
- ROCKWELL, D. 1977, Prediction of Oscillation Frequencies for Unstable Flow Past Cavities, *Journal of Fluids Engineering*, 99, 294-300.
- ROSHKO, A. 1961, Experiments on the Flow past a Circular Cylinder at Very High Reynolds Number, *Journal of Computational Physics*, 43, 345-356.
- STAM, J. 1999, Stable Fluids, In *Proceedings of ACM SIGGRAPH 99*, Annual Conference Series, 121-128.
- STROUHAL, V. 1878, Ueber eine besondere Art der Tonerregung, *Ann. Phys. Chem. (Wied. Ann. Phys.)*, 5, 216-251.
- TAKALA, T. AND HAHN, J. 1992, Sound Rendering, In *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, 26, 2 211-220.
- TAM, C. K. W. 1995, Computational Aeroacoustics: Issues and Methods, *American Institute of Aeronautics and Astronautics Journal (AIAA Journal)*, 33, 10, 1788-1796.
- TSINGOS, N., FUNKHOUSER, T., NGAN, A., AND CARLBOM, I. 2001, Modeling Acoustics in Virtual Environments Using the Uniform Theory of Diffraction, In *Proceedings of ACM SIGGRAPH 2001*, Annual Conference Series, 545-552.
- VAN DEN DOEL, K., KRY, G., AND PAI, D. K. 2001, Foley Automatic: Physically-based Sound Effects for Interactive Simulation and Animation, In *Proceedings of ACM SIGGRAPH 2001*, Annual Conference Series, 537-544.
- VAN DEN DOEL, K., AND PAI, D. K. 1998, The Sounds of Physical Shapes, *Presence: Teleoperators and Virtual Environments*, 7, 4, 382-395.
- WILLIAMSON, C. H. K. 1988, The Existence of Two Stages in the Transition to Three-Dimensionality of a Cylinder Wave, *Physics of Fluids*, 31, 11, 3165-3168.

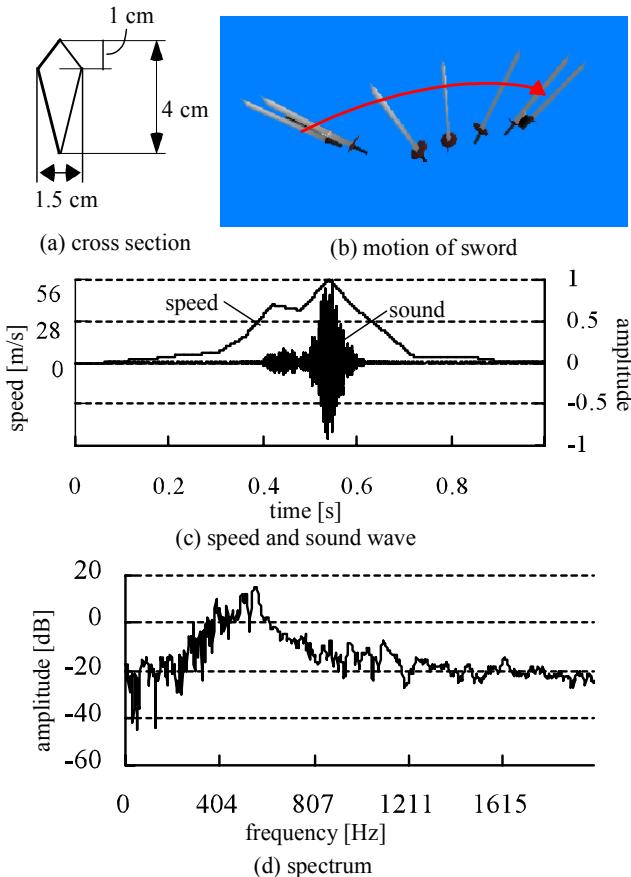


Figure 9: Aerodynamic sound by swinging a sword.

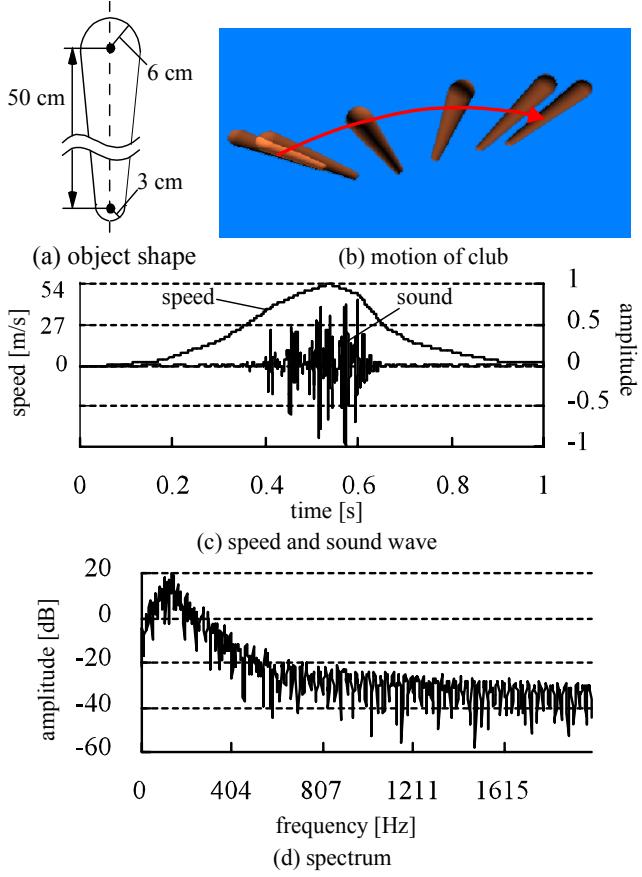
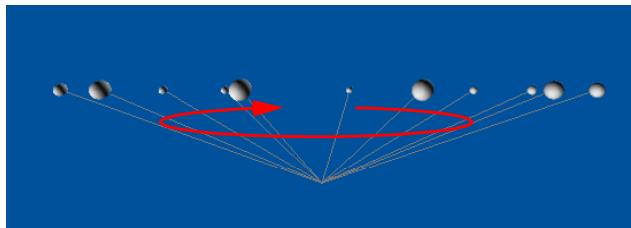
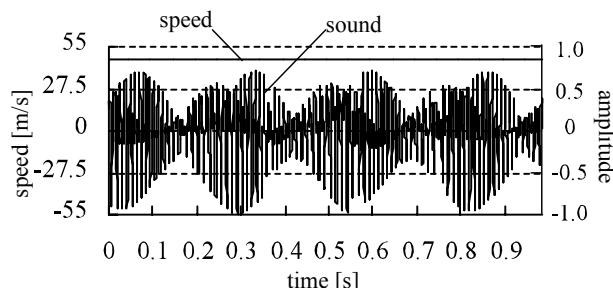


Figure 10: Aerodynamic sound by swinging a club.

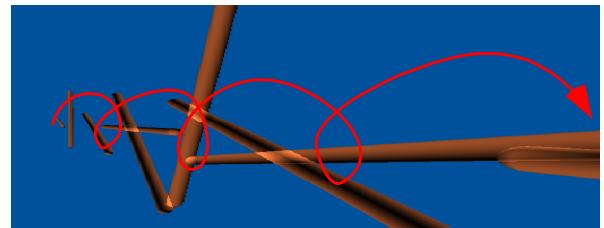


(a) motion of sphere

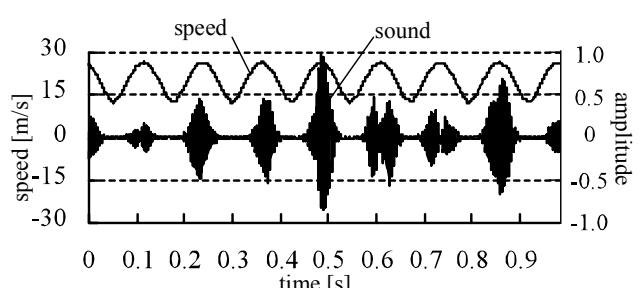


(b) speed and sound wave

Figure 11: Aerodynamic sound by rotating sphere. The radius of the sphere is 5 cm.



(a) motion of circular cylinder



(b) speed and sound wave

Figure 12: Aerodynamic sound by circular cylinder approaching the receiver. The radius and length of the cylinder are 1 cm and 50 cm, respectively. In Fig. (a), the speed is measured at an end point of the cylinder.

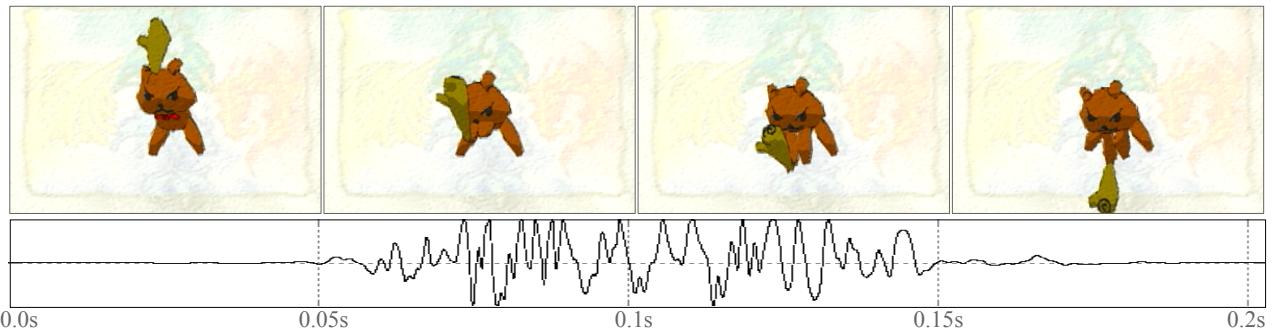


Figure 13: Example of character animation. A bear swings a huge club.

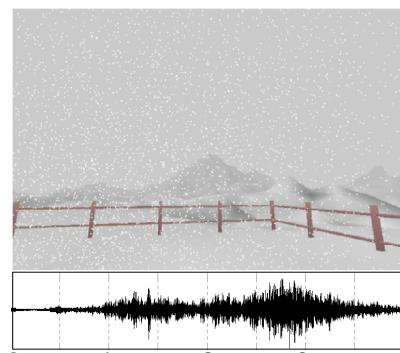


Figure 14: Aerodynamic sound by a cold wintry wind.

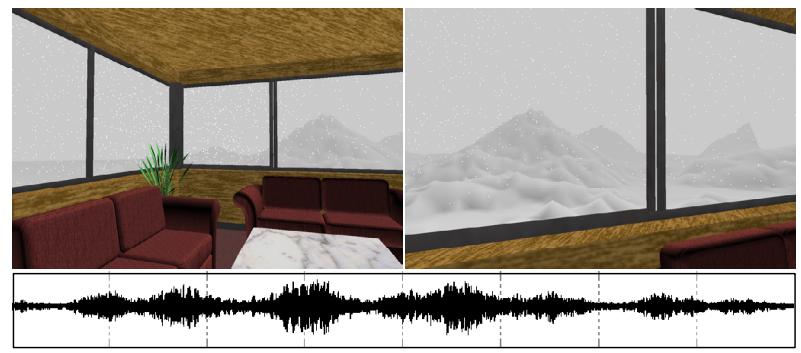


Figure 15: Aerodynamic sound by a draft. The left image shows the entire room and the right image shows the gap between the windows.