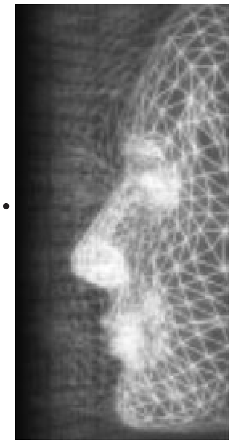
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2005; **16:** 531–545
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cav.58

Generation of tree movement sound effects



By Katsutsugu Matsuyama, Tadahiro Fujimoto, Kazunobu Muraoka and Norishige Chiba*

This paper presents a method for automatically generating sound effects for an animation of branches and leaves moving in the wind. Each tree is divided into branches and leaves, and an independent sound effect generation process is employed for each element. The individual results are then compounded into one sound effect. For the branches, we employ an approach based on the frequencies of experimentally obtained Karman vortex streets. For the leaves, we use the leaf blade state as the input and assume a virtual musical instrument that uses wave tables as the sound source. All computations can be performed independently for each frame step. Therefore, each frame step can be executed on completion of the animation step. The results of the implementation of the approach are presented and it is shown that the process offers the possibility of real-time operation through the use of parallel computing techniques.

Received: April 2003; Revised: April 2004

KEY WORDS: computer animation; natural phenomena; moving trees; sound modelling; sound rendering

Introduction

Sound effects are used in a wide range of applications including animations, movies and computer games. In the conventional generation process, a sound designer manually creates and adds the sound effect based on the video images. This is a potentially complicated and time-consuming method that may necessitate an expert sound designer. In interactive contents such as virtual reality (VR) and mixed reality (MR), the cue video is dynamic, so there is a significant demand for automatic sound effect generation techniques. Research on automatic sound effect generation has been conducted in the field of computer graphics[1] and has been increasingly reported on in the last 2–3 years.[2–12] As with modelling and rendering for computer graphics, sound-generating techniques for virtual three-dimensional environments can be broadly divided into sound-modelling and sound-rendering methods. Sound modelling uses the shape and properties of an object to construct the waveforms that the object emits as a sound source, while sound rendering generates a filter as the sound effect computed from the echoes reaching the listening point from the sound source.[1]

The method proposed by Doel *et al.*[2] automatically adds sound effects for collisions, friction and rotation based on physical laws. This pioneering approach automatically and interactively adds sound effects. It has now become possible to create natural sounds by measuring the sounds of actual 3D objects.[13] One technique creates sounds using the infinitesimal changes in geometrical shape over a short epoch.[3] This uses a nonlinear finite element method[14] that enables automatic generation of physically persuasive and natural sound effects, although the computational cost is large. A technique has also been proposed for reconstructing sounds by performing wavelet analysis in a static manner and projecting onto the character or texture.[4] Dobashi *et al.* proposed an approach for aerodynamic sound using sound textures developed independently of ours but which has some similarities.[5]

Sound-rendering research has yielded a technique using a beam-tracing method[6] whereby the user can move interactively with the sound source fixed by emitting beams from the sound source. This approach has been improved variously, and a method for sound rendering between avatars was proposed that allows

*Correspondence to: Katsutsugu Matsuyama, Department of Media Architecture, Future University-Hakodate, 116-2, Kameda, Nakanocho, Hakodate, Hokkaido, 041-8655, Japan
E-mail: kmatsu@fun.ac.jp

Contract/grant sponsor: Science and Technology Research Program of Iwate Prefectural Government.

free movement by executing an improved beam-tracing method in real time.[7] This method has also been modified with changes to the beam-tracing[8] and diffraction[9] elements. A sound-rendering method that operates in real time has been proposed in the field of geometrical acoustics.[10] Calibration of the indoor acoustics and verification of the various beam-tracing methods were performed using the Bell Labs Box auditory environment. The approach is based on the image source method. With this method, it is possible to position and listen to an orchestra in a virtual 3D environment. However, such interactive applications have already been presented.[11] One study reports on the creation of an artistic environment by constructing an interactive musical stage environment,[12] while Cook[15] provides a detailed discussion of methods related to computer graphics and describes sound-related physical features and general techniques.

Our research group has developed a noise-based efficient technique to generate a real-time animation of wind-induced tree branch and leaf movement (see Figure 1a), and this paper proposes a sound-modelling method that automatically adds sound effects representing the wind-induced movement (see Figure 1b). Details of the wind-induced waving animation technique will be released on another occasion because they are beyond the scope of this paper. It is conceivable to use the sounds recorded from actual trees to generate the required effects. However, it is considered difficult to process or synthesize the sound to match the video if, in particular, the video is determined dynamically. Therefore, the new approach provides an automatic technique that recognizes the individual leaves and branches in the animation. In our method, the sound effect for a tree is generated by estimating the contribution of each branch and leaf so that the natural sound

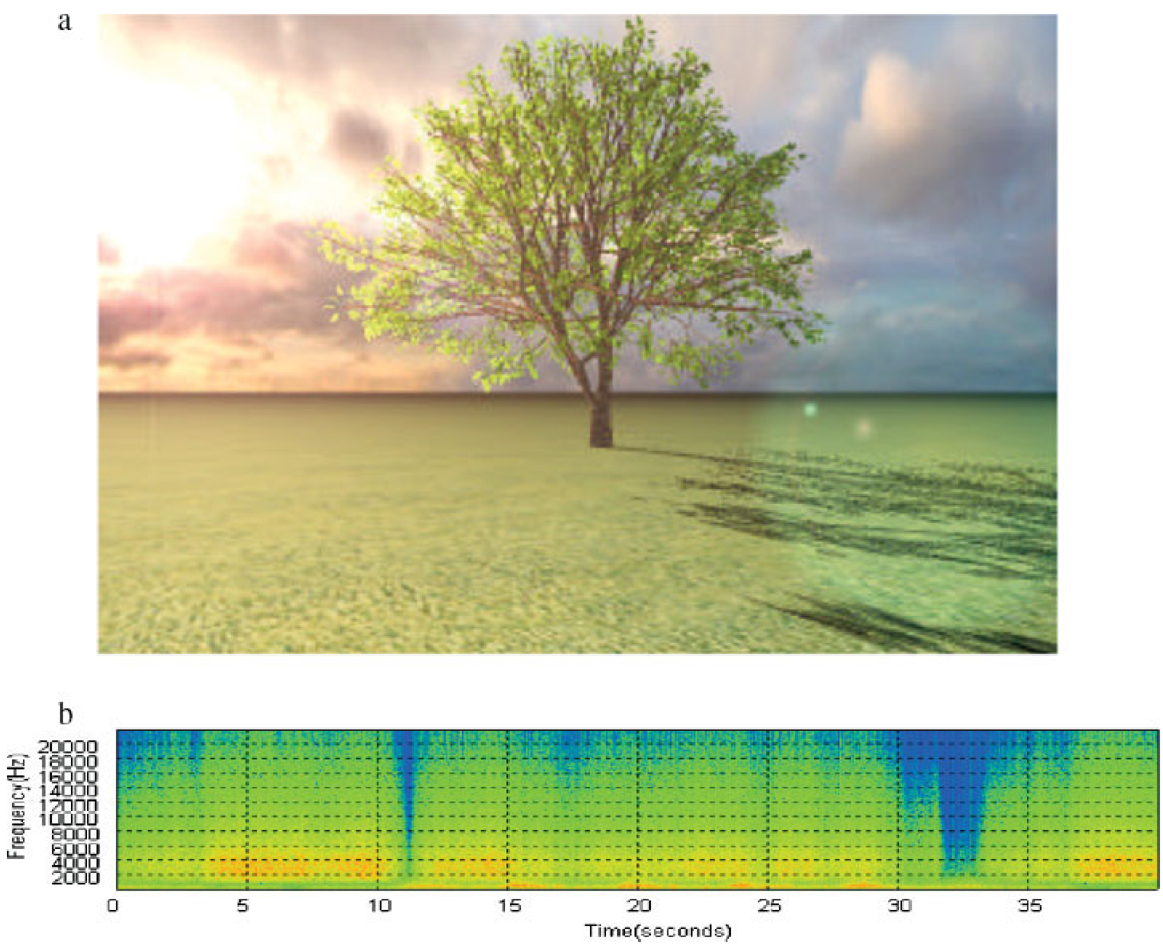


Figure 1. Sound effect generation: (a) a waving tree; (b) generated sound spectrum.

source distribution according to the structure of the tree that is distinguishable in a close-range view is realized.

Sounds are waves propagated in a medium, and this paper assumes the medium to be air. Sound waves have the important properties of period, waveform and amplitude, and people judge the type of sound source by distinguishing these as pitch, tone and loudness. These are called the three elements of sound and they are each subject to change over time. The generation of a sound can be paraphrased as the determination of these three elements. For example, a synthesizer or MIDI instrument can be divided into a sound source block and control block, the former handling the waveform and the latter handling the pitch and loudness.

The system proposed in this paper is shown in Figure 2. The approach can be divided into branch sound generation and leaf sound generation processes, and these can be treated as independent except when referring to the branch shape data during leaf movement. In the branch sound-generating process, branch movement shape data is generated from the branch shape data and the noise function representing the wind. The branch sound is generated from the shape data and the noise function representing the wind. Leaf sound is generated using the leaf movement shape data and adjacent relational data.

The branch movement and branch sound-generating process are described in the next section, and leaf movement, pre-processing and leaf sound generation processing are described in the fourth section. In the next two sections, the sound effects of a tree moving in the wind are described. As this is sound created from the sounds emitted by the tree itself, the tree can be treated as the sound source. Using this approach in the sound-rendering method described in the fifth section, the tree sound effect is created in a virtual environment. Execution results and related experiments are described in the sixth section, and the seventh section briefly summarizes the study and discusses future research.

Branch Sound Generation Process

To develop a technique to automatically generate tree sound effects, it is necessary to implement a tree shape model and the associated movement generation method. For the tree shape model, various existing techniques can be applied.[16–18] For the generation of wind-induced tree branch and leaf movement, an efficient noise-based technique developed by the authors is used. This section explains the technique for generating the sound of the tree branches.

Branch Shape Model

In the branch model, the branch internodes are approximated as cylinders or truncated circular cones. The data for the entire tree is tree-structured and the parent and child internodes are linked as shown in Figure 3. This simplifies the process of obtaining the accumulated movement of the individual branches. If a branch has

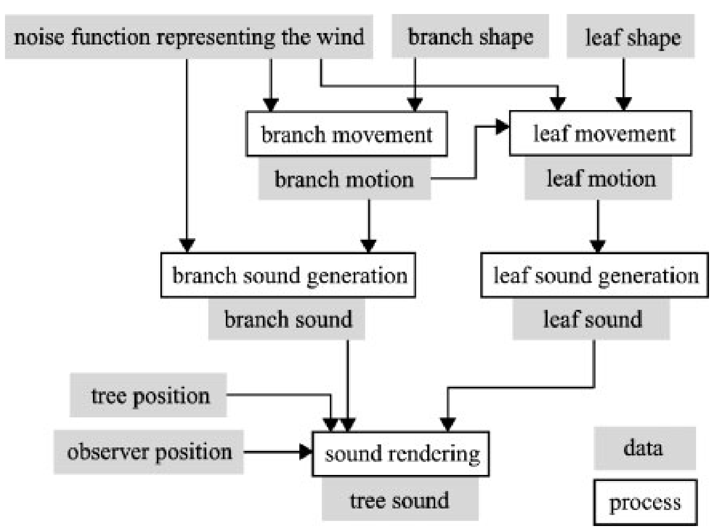


Figure 2. Overview of proposed system.

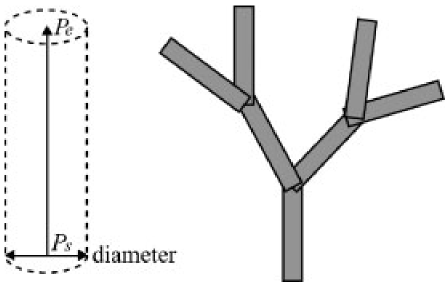


Figure 3. Tree branch model.

no child, it is a cylinder, and if it has a child, it is a truncated circular cone with the diameters of that branch and the child. Each branch is defined by the diameter and vector from the root to the branch tip with start point P_s and end point P_e as shown in Figure 3. This is all the data that is required in relation to the branch shape for branch sound effect creation.

Branch Movement

To generate a branch sound, a vector representing the wind and branch velocities is required. Here, the branch velocity is a vector that represents the positional change of points P_s and P_e during each epoch. In our study, branch movement is represented by a noise-based real-time simulation of tree movement due to wind (see Figure 4). This means that the wind-induced load is represented by applying $1/f^{\beta}$ noise to the spring model of a cantilever. In $1/f^{\beta}$ noise, the spectrum density at frequency f in the frequency region has a magnitude in proportion to $1/f^{\beta}$. The β value determines the correlation of noise along the space–time axis.[19,20] This noise is observed in many fluctuating phenomena in nature.[21]

We take the x- and y-axes in a plane perpendicular to the branch with the origin at the lower branch node, as shown in Figure 5, and show the movement amplitude on both axes. The deflection angle when load P is imposed on the branch is as follows:

$$\delta_x(t) = \frac{P_x(t)}{k} \tag{1}$$

$$\delta_y(t) = \frac{P_y(t)}{k} \tag{2}$$

The spring constant k depends on the modulus of elasticity E, where E is unique to the tree species and does not depend on the length or thickness of the branch. For each branch, let the loads $P_x(t)$ and $P_y(t)$ in the x- and y-axis directions be as follows:

$$P_x(t) = F_x(t) + P_B N_x(t) \tag{3}$$

$$P_y(t) = F_y(t) + P_B N_y(t) \tag{4}$$

where $F_x(t)$ and $F_y(t)$ represent the directional loads given by wind, and P_B is the maximum load. $N_x(t)$ and $N_y(t)$ are $1/f^{\beta}$ noise functions and represent non-directional loads.

The amplitude of the movement of each branch is the accumulated amplitude from the tree root to the child node, as shown in Figure 6. Thus, the branch movement of the entire tree is determined. By performing this process in frame steps, the branch state is determined for each frame.

Figure 4. A deciduous tree in winter.

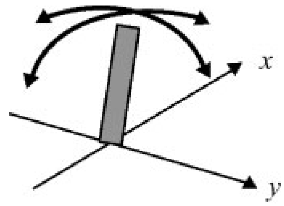


Figure 5. Branch waving in the x- and y-axis directions.

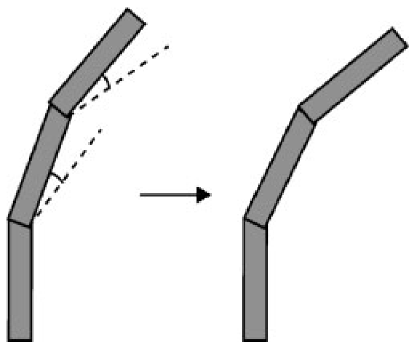


Figure 6. Accumulated waving amplitude.

Generation of Branch Sound

We generate branch sounds from the directional wind obtained in the previous subsection and the branch movement due to the wind. The branch sound effect is actually a vortex-emitted sound. The frequencies of Karman vortex streets were obtained experimentally,[22] and the wave tables are created on the fly using the obtained frequency. The technique for adding the sound effect is based on these frequencies for a sound emitted from a cylinder of diameter D in flow of velocity U.

(I) Frequency Model for Karman Vortex Street. The frequency f of a Karman vortex street can be described by the following formula:[22]

$$f = S_t U/D \tag{5}$$

S_t : Strouhal number

The Strouhal number is a function of Reynolds number Re and its value is almost constant at 0.2 if Re is in the range $4 \times 10^2 < Re < 2 \times 10^5$. Therefore, the proposed method treats the Strouhal number as a constant of 0.2. This formula indicates that the sound frequency rises in proportion to the flow velocity and in inverse proportion to the diameter. The sound intensity I is

$$I \propto U^6 \cdot l/r^2 \tag{6}$$

l : Length of the cylinder

r : Distance from branch to observer

The vortex-emitted sound is actually a directional dipolar sound. However, the proposed method treats it as having no directional dependence for simplicity.

To find the flow velocity, we determine the relative velocity between the wind and branch and generate a sound using the relative velocity as a parameter. Therefore, using the branch velocity and wind direction and intensity for each epoch, we calculate the component of the velocity perpendicular to the branch (Figure 7).

The wind intensity is related to the movement by the loads on each branch represented by $F_x(t)$ and $F_y(t)$ in formulae (3) and (4), and is also implemented as $1/f^{\beta}$ noise. A constant wind can be represented by adding a time-independent component to this noise.

We apply formulae (5) and (6) at points P_s and P_e of each branch to compute the frequency and sound intensity. These are the values at the tips of the branches. To create the frequency group for an entire branch, let F_s and F_e be the frequencies obtained at P_s and P_e and let I_s and I_e be the intensities at these points. We determine the frequencies and sound intensity of the entire branch by linear interpolation (Figure 8).

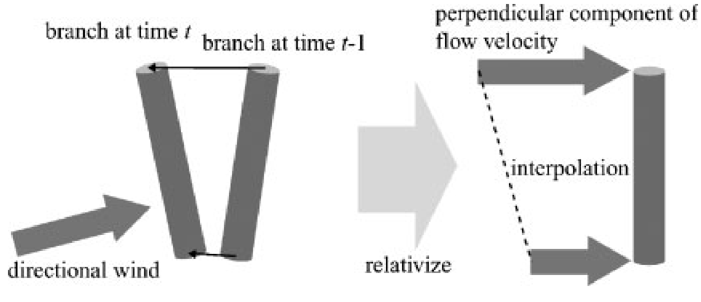


Figure 7. Relative velocity.

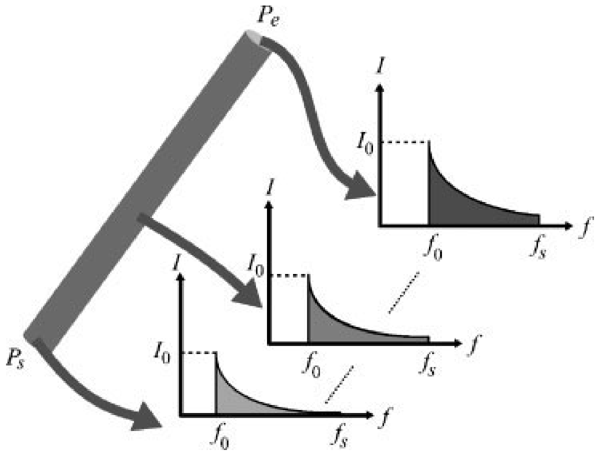


Figure 8. Creation of frequency group.

We represent the obtained frequencies in complex conjugate form and assign a random value for each phase; then we generate the sound along the time domain by inverse fast Fourier transformation. Branch sound between frames can be generated by performing this in frame steps. If all the phases are the same, a noticeable beat will appear. This can be eliminated by assigning a phase difference using random numbers when disposing in the complex frequency region.

If branch sounds between frames are directly coupled, a change of frequency will be accompanied by a discontinuity of the waveform. This will be sensed as a keen pulse-shaped noise. Therefore, we generate a smooth signal by multiplying the signal by a window function to smoothly attenuate the signal at both extremes. A Gaussian function is used for the window function:

$$\mathrm{Gaussian}(x) = ce^{iax}e^{\frac{-(x-b)^2}{4\alpha}} \tag{7}$$

It is necessary to know the branch state several frames ahead. This is solved by computing several frames ahead of the imaging.

(2) Determination of Tone. We have described the technique for obtaining the frequencies and sound intensity of the branch movement, so it will become possible to add the sound once the tone is determined. In this paper, f_0 is taken to be the frequency obtained in subsection (1) and I_0 the intensity. High-frequency sound components are added in the range $f_0 < f_i < f_s/2$ using the following formula:

$$I_i = I_0/(f_i/f_0)^{\alpha} \qquad f_0 < f_i < \frac{f_s}{2} \tag{8}$$

α : Constant value which determines a tone

f_s : Sampling rate

where f_i represents the fineness, and a higher density of f_i will give a sound of higher quality (see Figure 8). However, the density may be determined according to the computing power of the CPU.

Generation of Leaf Sound

In the previous section, we presented the method for automatically adding sound effects for the movement of tree branches, a method that can be operated as an independent system. In this section, we describe a method for creating sound effects for leaf movement (see Figure 9).

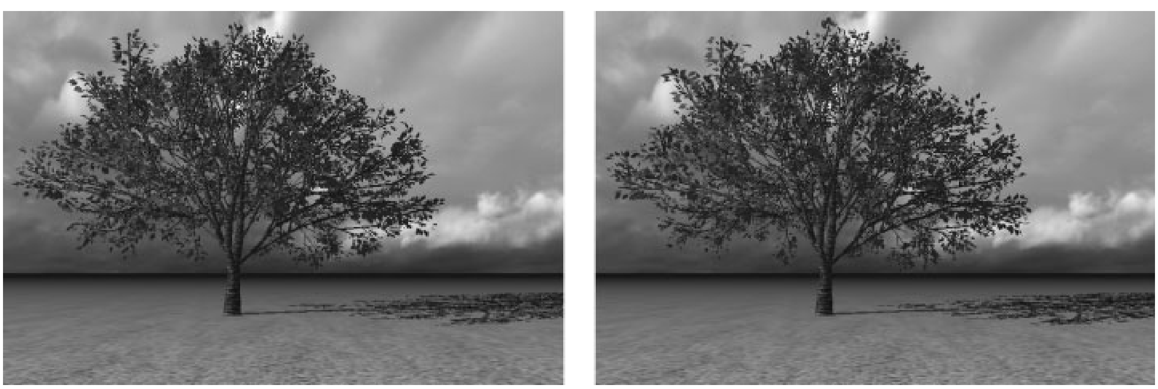


Figure 9. A deciduous tree in summer.

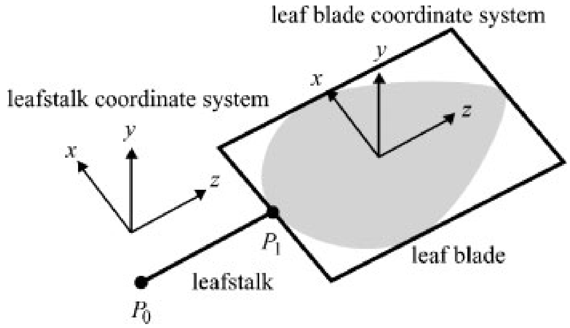


Figure 10. Leaf shape model: leaf stalk coordinate system and leaf blade coordinate system.

Leaf Shape Model

In the proposed leaf model, leaf stalks are approximated by a line segment and the leaf blades by a rectangle, as shown in Figure 10. The leaf blade is created based on leaf stalk data and its size is proportional to the length of the leaf stalk. A leaf stalk coordinate system is defined to represent leaf movement, with the origin at point P_0, and the vector from points P_0 to P_1 is the z-axis. The normalized cross product of the z-axis and a vertical upward vector is the x-axis, and the normalized cross-product of the z- and x-axes is the y-axis. The leaf stalk coordinate system is static. The movement of the leaf is described in the leaf stalk coordinate system. The leaf blade droops at the contact between the leafstalk and leaf blade due to gravity. Therefore, point P_0 does not lie on the plane of the leaf blade.

For leaf sound generation, it is necessary to know the relative positional change of each leaf blade with time. Therefore, we define the leaf blade coordinate system from the leaf blade position determined in each frame. This is determined dynamically. We take the origin at the centre of the rectangle that represents the leaf blade. Let the y-axis be in the direction normal to the rectangle and the z-axis be towards the tip of the leaf blade. Take the normalized cross-product of the y- and z-axes as the x-axis. For a leaf blade shape that is curved from point P_1 toward the leaf tip, we determine the leaf blade plane by differentiating the curved surface at its centre and define the leaf blade coordinate system in that plane in a similar manner.

Representation of Leaf Movement

To represent leaf movement, we employ a method similar to that for branch movement. This method generates noise-based movement for each leaf. Actual wind-induced tree leaf movement involves complex elements such as leaf size and shape, leaf stalk length and thickness, leaf and leaf stalk stiffness, and the wind characteristics. In addition, a vast number of leaves must be processed. The proposed method realizes a technique to efficiently represent such complex movement. We divide the leaf movement into longitudinal and lateral movements and a rotational movement around the leaf stalk axis, and assign a movement or rotation amplitude to each using noise. The lateral and longitudinal movement directions are the x- and y-axis directions defined in the branch shape model. Let $\theta_x(t)$ and $\theta_y(t)$ represent the movement amplitudes at time t in the respective axis directions and $\theta_{rf}(t)$ represent the rotation amplitude in the rotational direction about the z-axis. These are determined by the following formulae:

$$\theta_x(t) = W_x N_x(t) \tag{9}$$

$$\theta_y(t) = W_y N_y(t) \tag{10}$$

$$\theta_{rf}(t) = W_r N_r(t) \tag{11}$$

where W_x and W_y represent the maximum waving amplitudes, W_r represents the maximum rotation amplitude, and $N_x(t)$, $N_y(t)$ and $N_r(t)$ represent noise functions. The maximum amplitudes determine the range over which the leaf moves and serve as parameters to represent the wind velocity and various tree species. For the rotation amplitude, consider the case where the leaf moves laterally because such movement is accompanied by a rotation about the leaf stalk axis. Thus, the torsion angle is defined by the following formula using the x–r coupling coefficient a:

$$\theta_{rx}(t) = a\theta_x(t) \tag{12}$$

As a result, the leaf rotation amplitude θ_r is

$$\theta_r(t) = \theta_{rf}(t) + \theta_{rx}(t) \tag{13}$$

Each leaf is first given a movement by applying a noise function with a different phase. Then the leaf movement over an entire tree is represented by integrating each leaf movement with the corresponding simulated branch movement.

Leaf Sound Generation

Leaf sound is generated by the collisions and friction between the leaves. The output data obtained from the leaf movement algorithm is used to represent the position and posture of the leaves at time t. The time is represented in frame steps. Therefore, if the reproduction frame rate is taken to be, for example, 30 fps and the reproduction audio sampling rate 44100 Hz, then the period from time $t-1$ to time t, or one frame, is $1/30$ second, and $44100/30$ or 1470 audio samples will exist. Here, using the leaf position and posture data at time t and $t-1$, we compute the sound generated from the leaf blades during this frame. We propose a technique to generate sound effects using a virtual musical instrument, which will use the leaf state as an input and the wave tables as a sound source. This means creating a virtual synthesizer, which uses tones of the three elements of sound as a sound source, and assigning it to a leaf. Frequency and sound intensity are inputs and are computed from the motion of the leaf. All the leaf blades can be sound sources and can also act as inputs. For example, if leaf L_j influences leaf L_i, sound generation takes place using leaf L_i as a musical instrument and the positional relationship between the leaves as the input. To find the positional relationship, it is necessary to create a leaf blade coordinate system for leaf L_i and compute the position of leaf L_j in that coordinate system. In this paper, we perform efficient sound effect generation without strict collision judgment. To generate a leaf sound, it is necessary to search for leaves in the neighbourhood of the leaf under consideration. However, for every leaf, our program creates a list of leaves existing within a certain distance as pre-processing and uses these lists instead of searching.

(I) Creation of WaveTables. Each leaf blade forms a sound source and the tone is determined by creating wave tables, but a vast amount of memory will be consumed if every leaf is given a wave table. Therefore, we employ a technique for creating a number of wave tables first and assigning one of the wave tables to each individual leaf according to the leaf blade size (Figure 11). As the wave tables are reproduced directly, we must be attentive to the waveform. In fact, it is thought that two wave tables for collision and for friction should be assigned to each leaf. We suggest that the wave tables for collision and friction both depend on the leaf blade size, and employ a technique that uses one wave table. In this technique, collision and friction are distinguished by altering the reproduction speed of the wave table. In this paper, a wave table is synthesized by using a pink noise and adjusting its basic frequency. In this study, the basic frequency was chosen by trial and error.

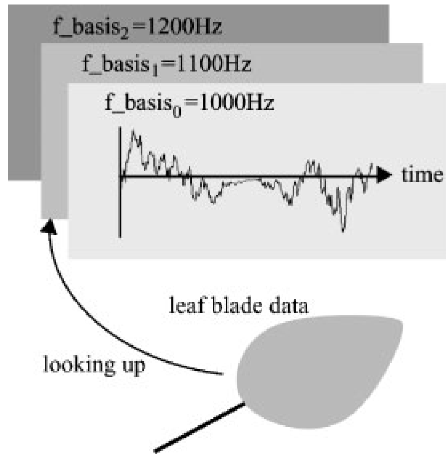


Figure 11. Wave table.

This process forms a sound source and reproduces a sound when a collision or friction signal is entered. We use a 'stylus' to represent the reproducing position and initial value of the current wave table. This corresponds to a record stylus, and sound is reproduced as the stylus proceeds. By this process, continuous reproduction of the signals is made possible. For friction, the relative leaf velocity is used as an input and the sound-reproducing speed is altered according to the relative leaf velocity. A higher sound can be reproduced for a higher leaf velocity in such a manner. The initial stylus state is set to the zero-crossing point of the waveform. This is to prevent pulse-shaped noises from occurring even for steep input signals.

The sound quality will be improved if the number of wave tables is increased. Therefore, using as much memory as possible improves the sound quality but increasing the number of wave tables requires a longer computation time. As this can be regarded as pre-processing for the sound effect generation process, a better result can be expected with more wave tables. However, the wave tables not referred to are redundant. For a large tree, a sound effect that is natural to the senses cannot be obtained unless many wave tables are secured.

For wave table creation, it is appropriate to increase the basic frequencies of the wave tables in equal increments. However, exponential increments are more natural to the senses.[23] Let fm be the minimum basic frequency, and consider placing N wave tables in one octave from there. In this case, set each basic frequency f_basis_i according to the following equation:

$$f\_basis_i = fm^* 2^{i/N} \tag{14}$$

Such an assignment will generate a sound that is more natural to the senses. In this case, care must be taken when assigning wave tables to the leaves.

(2) Control of Sound Sources. A wave table is assigned to leaf L_i in subsection (1). Next, the behaviour of leaf L_j must be computed in the leaf blade coordinate system of leaf L_i. With this in mind, we compute the positional change of leaf L_j in the leaf blade coordinate system. Take the centre of the leaf blade as a representative point of leaf L_j, and denote it by P_j. Both leaves L_i and L_j vary their position over time. Describe leaf L_i at time t as leaf $L_i(t)$ and point P_j at time t as point $P_j(t)$. Then, a conversion to the leaf blade coordinate system can take place as an inner product:

$$\begin{aligned} Lv_x &= VL_i(t)_x \cdot (P_j(t) - P_i(t)) \\ Lv_y &= VL_i(t)_y \cdot (P_j(t) - P_i(t)) \\ Lv_z &= VL_i(t)_z \cdot (P_j(t) - P_i(t)) \end{aligned} \tag{15}$$

where $VL_i(t)_k$ is the k-axis vector of the leaf blade coordinate system $VL_i(t)$ of the leaf $L_i(t)$ described in the global coordinate system, and $Lv = (Lv_x, Lv_y, Lv_z)$ is the position vector of $P_j(t)$ described in $VL_i(t)$. We regard this relation between leaves as a state and compare the distance from the leaf blade coordinate system origin of leaf L_i to point P_j against the threshold. A state transition is then performed according to the results of the comparison and the threshold established according to the leaf blade size. We consider five states for each leaf: contact state, non-contact state, Key on, Key off and initial state. A state transition diagram is shown in Figure 12. Each state is described below.

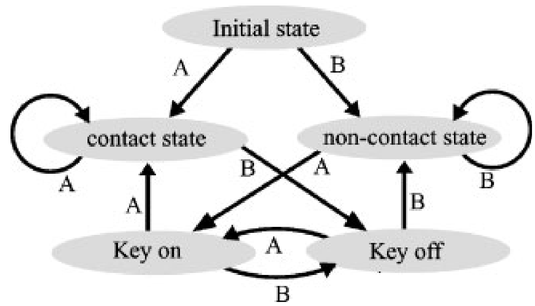


Figure 12. Leaf state transition diagram. A: Distance from leaf blade coordinate system origin of leaf L_i to that of leaf $L_j \leq$ threshold. B: Distance from leaf blade coordinate system origin of leaf L_i to that of leaf $L_j >$ threshold.

(a) *Contact state*. This represents the state of leaf blades being in frictional contact. The wave table is reproduced according to the friction speed. If the distance between the leaf blade coordinate systems is small, the leaves are interpreted as maintaining contact and make no state transition. Otherwise, the transition is to the Key off state.
(b) *Non-contact state*. This represents the state of leaf blades not being in contact. In this case, no action occurs because the leaf blades are distant. A transition to Key on takes place if the distance from the origin of the leaf blade coordinate system is small. Otherwise, no transition takes place.
(c) *Key on*. This represents the transitional state of leaf blades being in collision. We compute the collision time using the leaf positions at time t and at time $t-1$ and threshold and raise an envelope using the collision speed as shown in Figure 13. If the distance from the origin of the leaf blade coordinate system is small, the leaves are interpreted as maintaining contact and transition to the contact state. Otherwise, the transition is to the Key off state.
(d) *Key off*. This represents the transitional state of leaf blades having ended collision or friction. By assuming this state, we prevent pulse-shaped noise when the sound waveform rapidly decays to zero,

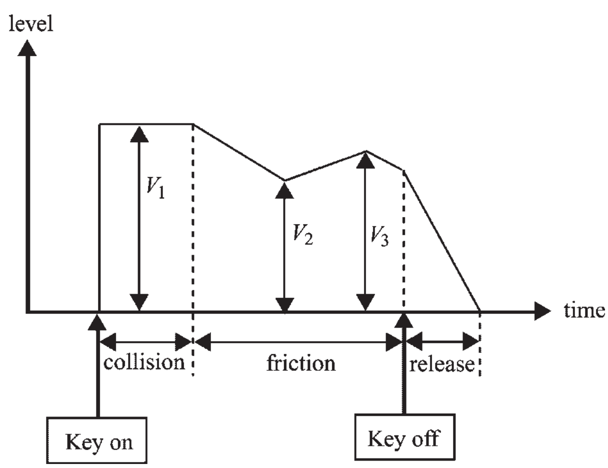


Figure 13. Example of envelope.

that is, the decay will take a long period of time. If the distance from the origin of the leaf blade coordinate system is small, the leaf makes a transition to Key on at the next collision. Otherwise, the transition is to the non-contact state.

(e) *Initial state*. This makes a transition to the contact or non-contact state according to the distance from the origin of the leaf blade coordinate system.

State transition will take place in each frame step. Here, we want to send a Key on message, in audio steps, to the sound generated when a transition takes place to the Key on state. Therefore, it is transformed into audio steps and sent as shown in Figure 14 according to the state of the frames before and after. When the initial state makes a transition to the contact state, the start position of the audio steps is determined using random numbers. The sound intensity is assumed to depend on the velocity for both collision and friction, and it is determined from the positional change of point P_j in the leaf blade coordinate system.

Sound Rendering

The technique described above makes it possible to automatically generate a sound effect for a tree moving in the wind. It is possible to treat the tree as the sound source because the process is the creation of the sound emitted from the tree itself. This section treats the tree as the sound source and generates the sound at a listening position from the source position and observer position and direction. Such a process is called sound rendering. This process is also called sound location in acoustic engineering and auralization in VR and other fields. Sound signal processors (for example, Roland RSS-10) with the sound localizing function can be used for the sound location and various techniques are available, including those using the sound-rendering methods proposed so far.[4,5,7,10] However, these proposed sound-rendering methods have been mainly conducted on indoor acoustics and the rapid and accurate identification of the sound path under complex boundary conditions. We used an outdoor environment in this study and decided not to consider reflection, refraction and the like at the boundary. Therefore, we do not use the above-stated complex sound-rendering methods but employ a sound generation method for the sound heard at the listening point. This is a technique to represent sound propagation.[1] It implements the delay, attenuation and directionality due to the distance and angle from the sound source to the observer and the changes accompanying the movement of the observer.

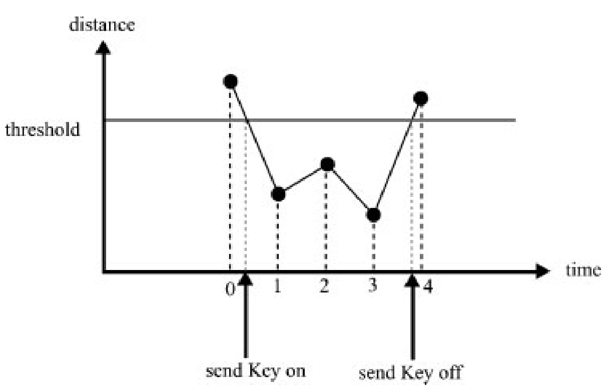


Figure 14. Transformation from frame steps into audio steps.

This technique places a unidirectional virtual microphone in a virtual space and records the tree sound. A cardioid function is available to represent the unidirectivity, and is expressed by the following formula:

$$S(\mu) = C + ((1 + \cos\mu)/2)^k \tag{16}$$

where μ is the angle between the direction of the vector (acoustic ray) from the tree toward the virtual microphone and the direction of the virtual microphone. We use $k = 1$ and $C = 0$ as factors to represent the directivity of the virtual microphone.

The speed of sound in air is approximately 340 m/s. Sound is received with a delay and attenuation according to the distance from the tree to the virtual microphone. Strictly speaking, the calculation must be performed by considering the distance between each branch and the listening point. However, the distances to all branches are approximated by the distance to the tree stem for faster processing. A model is employed to represent the delay for each frame. This model regards the sound emitted from the tree itself as an analogue record and reproduces the analogue record with a record stylus whose reproducing speed varies due to the rate of change of the delay. In this model, the Doppler effect can be obtained as an inevitable consequence. Conversely, sound decays in reverse proportion to the second power of distance. Considering this, the ultimate loudness is represented by the following formula using loudness A_0 at reference distance d_0:

$$A(d) = A_0(d_0/d)^2 S(\sigma)S(\mu) \tag{17}$$

where d is the distance from the sound source to the virtual microphone, σ is the angle formed by the sound source and sound ray, and μ represents the angle between the sound ray and virtual microphone to consider the directionality. The coefficients of $S(\sigma)$ are $k=0$ and $C=0$ because a non-directional sound source is assumed for simplicity.

Experimental Results

We implemented the above algorithm with C++, and executed it on a machine with a 2.53 GHz Pentium 4 CPU and 1.5 Gbytes RDRAM memory. We set the reproduction frame rate to 30 fps and the audio sampling rate to 44100 Hz. (See the animations of the experimental results on the web site: http://www-cg.cis.iwate-u.ac.jp/lab/gallery7-e.html. Figures 1a, 4 and 9 are excerpts from the animations.)

The example uses two types of tree data. Sound generation was performed for the following three cases: only branch sound, only leaf sound, and both sounds. The camera movement is set by plotting eye points and fixation points varying over time in the 3D space and interpolating them into spline curves. Figure 15 shows the branch sound spectrum (using logarithmic and ordinary axes), Figure 16 shows the leaf sound spectrum and Figure 17 shows the spectrum of the compound sound made by compounding the two together. One can feel the immersive spatial spread of sound source distribution at a listening position close to a tree. Checking the demonstration movies at our web site[24] might convince that the proposed method has an ability to generate considerably realistic sound effects.

Relation of Tree Data Size with Computation Time

In this section, we describe an experiment to investigate how the computation time varies with tree size. Two types of tree data were used. For each data set, we measured the computation time required for creating the sound per 30 frames of the branch sound-generating block and leaf sound-generating block. We measured the time with the branch sound and leaf sound-generating blocks divided into the control module and sound source module. The control module operates wave tables and the sound source module synthesizes wave tables. The spectrum density of the branch sound-generating block was set to maximum, the number of wave tables was 512 and the threshold set to 0.3 (m). The results are shown in Table 1.

These results indicate that the computation time used for the sound source module is dominant in both branch

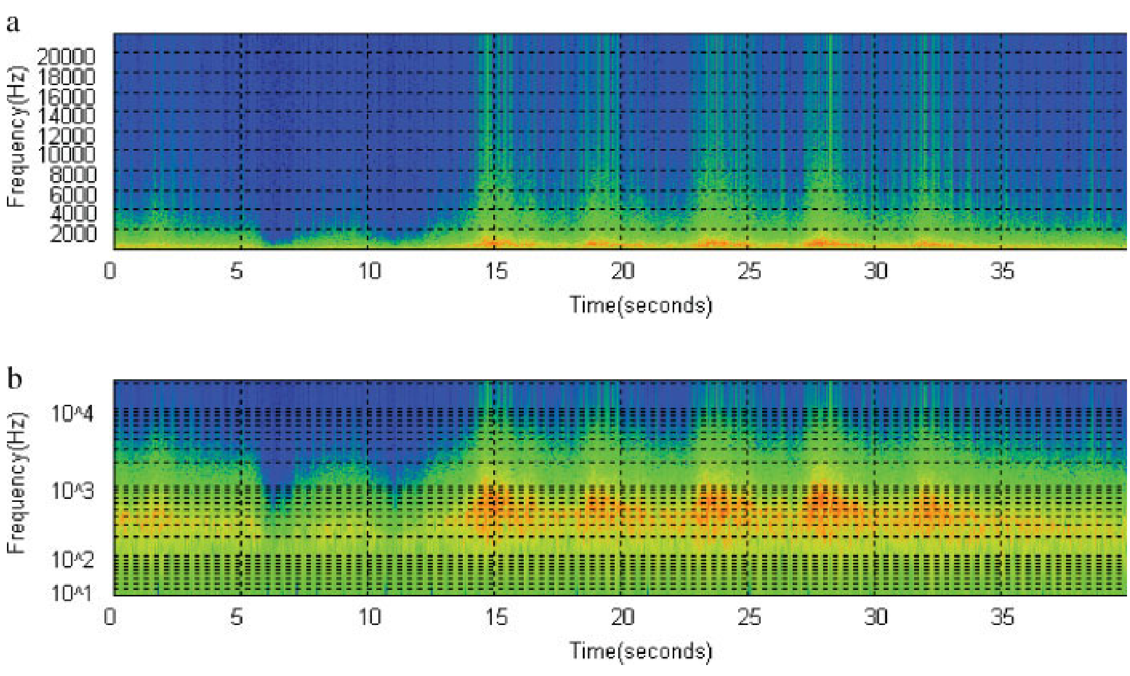


Figure 15. Branch sound spectrum. Top: ordinary axis. Bottom: logarithmic axis.

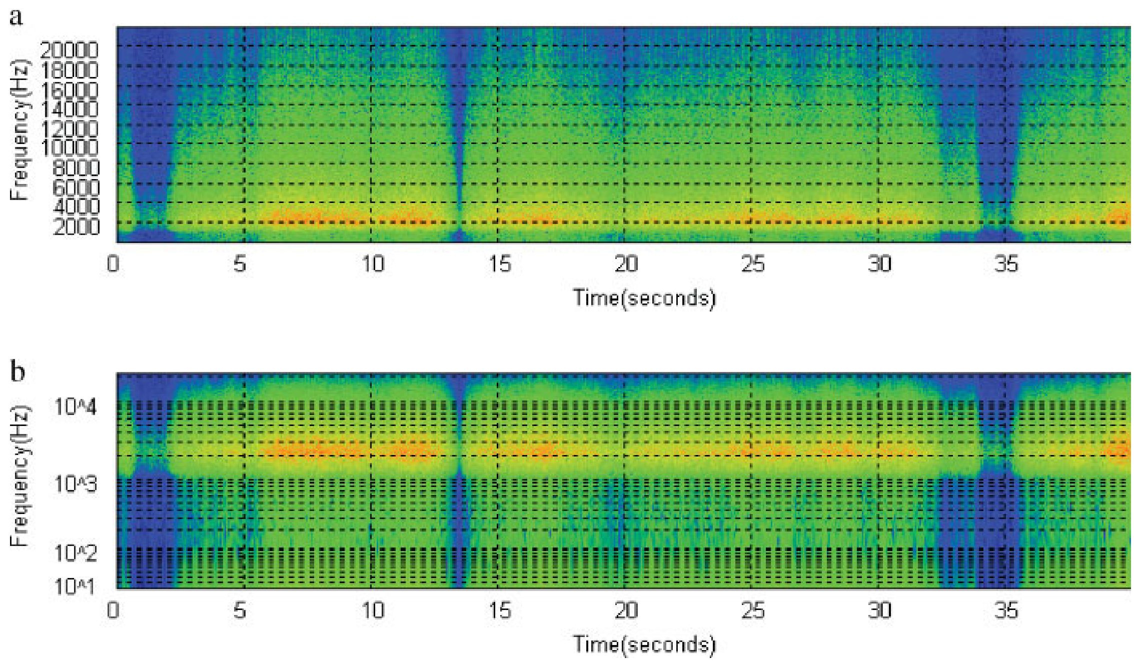


Figure 16. Leaf sound spectrum. Top: ordinary axis. Bottom: logarithmic axis.

sound and leaf sound. However, there is a possibility for faster processing because the system can be implemented as a parallel system with the sound source module as an independent function.

Experiment Results of Tone

For branch sound generation, as described in the third section, improved sound quality can be attained by

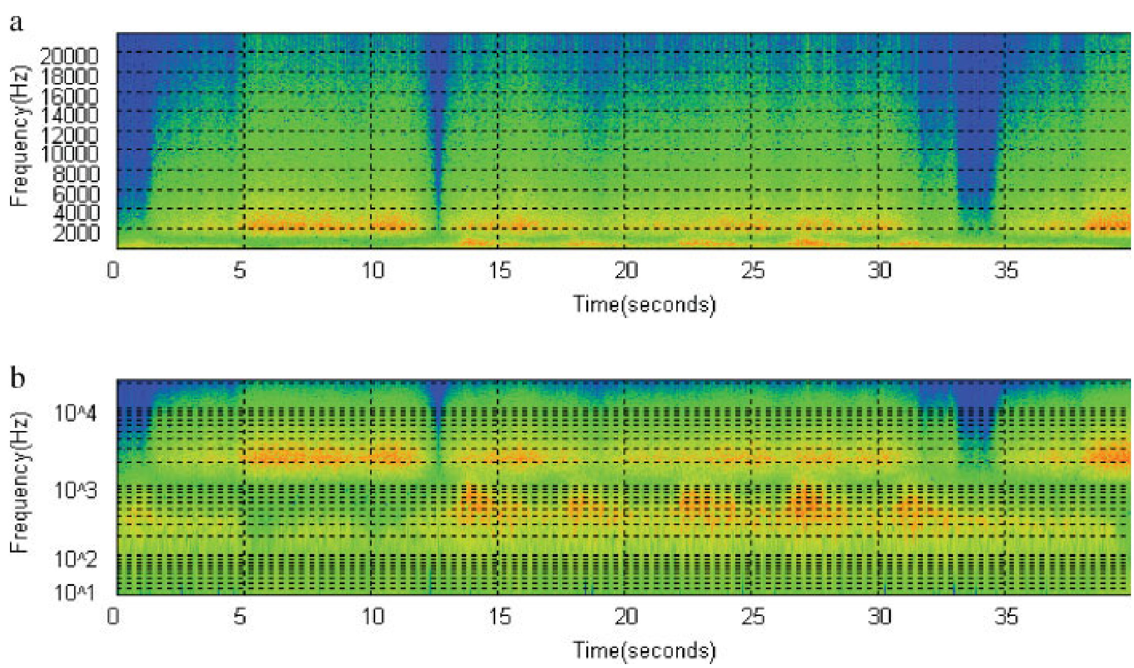


Figure 17. Spectrum obtained by compounding branch sound and leaf sound. Top: ordinary axis. Bottom: logarithmic axis.

(Branches, leaves)	Branch sound			Leaf sound		
	Total	Control	Source	Total	Control	Source
(123, 4860)	63.834	0.030	63.796	357.455	2.118	354.998
(1160, 16380)	578.208	0.201	577.954	645.524	2.743	642.205

Table I. Computation time (s)

using finer increments of f_i. If the increment is too coarse, the generated sound does not resemble actual sounds. As for the computation time, the decrease in computation time is relatively small compared to the deterioration in sound quality. This result indicates that a higher density is more desirable.

For leaf sound generation, the threshold is directly related to the computation time; that is, the amount to be processed per frame increases as the adjacent data increases. Therefore, appropriate threshold establishment is required. The number of wave tables does not depend on the amount of computation in terms of the time required for the sound-generating process; that is, more wave tables provide an improved sound quality. It is of course meaningless to increase the number of wave tables to the degree that there are wave tables that will never be referenced.

Summary and Future Research

This paper proposed a sound-modelling method to automatically generate a sound effect for an animation of tree branches and leaves moving in the wind. This technique divides the tree into branches and leaves, and we developed independent sound effect generation processes for both. The separate sound effects are then compounded into one sound effect. This paper provides a detailed description of the technique and the results of tests on the proposed system.

However, there are a number of aspects of the system that need further investigation. The generated sound effect must be assessed to see whether it closely resembles actual sounds, but there is no assessment technique that can be easily applied. Therefore, audiovisual assessment should be performed for a large number of subjects using statistical techniques.

The experiment results given in the previous section indicate that real-time generation is impossible with the present program. However, a higher efficiency can be expected by computing the branch and leaf sound sources independently in parallel, by parallel execution of leaf contact judgment, or by other means. Research on such a parallel computation method is expected.

Our technique considers only the sound emitted by a branch moving in the wind and the sound emitted due to the collision and friction between leaves. Besides these, a real tree generates sounds due to the collisions between branches and the collisions between leaves and branches, and the sound emitted when leaves are blown in the air. The effect of considering these factors remains to be examined.

We assumed a broad-leaf tree and have not examined coniferous trees. Additionally, we assumed wind intensity at ordinary levels, that is, we did not consider storm winds. These will be considered in future work.

ACKNOWLEDGEMENTS

This work was supported partly by the Science and Technology Research Program of Iwate Prefectural Government.

References

1. Takala T, Hahn J. Sound rendering. In *Proceedings of ACM SIGGRAPH '92*, ACM, Vol. 26(2), 1992; 211–220.
2. Doel K, Kry P, Pai D. Foley automatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of ACM SIGGRAPH 2001*, ACM, 2001; 537–544.
3. O'Brien J, Cook P, Essl G. Synthesizing sounds from physically based motion. In *Proceedings of ACM SIGGRAPH 2001*, ACM, 2001; 529–536.
4. Dubnov S, Joseph Z, Yaniv R, Lischinski D, Werman M. Synthesizing sound textures through wavelet tree learning. *IEEE Computer Graphics and Applications* 2002; **22**(4): 38–48.
5. Dobashi Y, Yamamoto T, Nishita T. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. In *SIGGRAPH 2003*, ACM, 2003; 732–740.

6. Funkhouser T, Carlbom I, Elko G, Pingali G, Sondhi M, West J. A beam tracing approach to acoustic modeling for interactive virtual environment. In *Proceedings of ACM SIGGRAPH '98*, ACM, 1998; 21–32.
7. Funkhouser T, Min P, Carlbom I. Real-time acoustic modeling for distributed virtual environments. In *Proceedings of ACM SIGGRAPH '99*, ACM, 1999; 365–374.
8. Min P, Funkhouser T. Priority-driven acoustic modeling for virtual environments. *Computer Graphics Forum* (*Eurographics 2000 Conference Proceedings*), Vol. 19(3), 2000; 179–188.
9. Tsingos N, Funkhouser T, Ngan A, Carlbom I. Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proceedings of ACM SIGGRAPH 2001*, ACM, 2001; 545–552.
10. Tsingos N, Carlbom I, Elko G, Kubli R, Funkhouser T. Validating acoustical simulations in the bell labs box. *IEEE Computer Graphics and Applications* 2002; **22**(4): 28–37.
11. Lokki T, Savioja L, Vaananen R, Huopaniemi J, Takala T. Creating interactive virtual auditory environments. *IEEE Computer Graphics and Applications* 2002; **22**(4): 49–57.
12. Reynolds M, Schoner B, Richards J, Dobson K, Gershenfeld N. An immersive, multi-user, musical stage environment. In *Proceedings of ACM SIGGRAPH 2001*, ACM, 2001; 553–560.
13. Pai D, Doel K, James D, Lang J, Lloyd J, Richmond J, Yau S. Scanning physical interaction behavior of 3D objects. In *Proceedings of ACM SIGGRAPH 2001*, ACM, 2001; 87–96.
14. O'Brien J, Hodgins J. Graphical modeling and animation of brittle fracture. In *Proceedings of ACM SIGGRAPH '99*, ACM, 1999; 137–146.
15. Cook P. Sound production and modeling. *IEEE Computer Graphics and Applications* 2002; **22**(4): 23–27.
16. Reffye P, Edelin C, Francon J, Jaeger M, Puech C. Plant models faithful to botanical structure and development. In *Proceedings of ACM SIGGRAPH '88*, ACM, Vol. 22(4): 1988; 151–158.
17. Chiba N, Ohshida K, Muraoka K, Miura M, Saito N. A growth model having the abilities of growth-regulations for simulating visual nature of botanical trees. *Computer and Graphics* 1994; **18**(4): 469–479.
18. Mech R, Prusinkiewicz P. Visual models of plants interacting with their environment. In *Proceedings of ACM SIGGRAPH '96*, ACM, 1996; 397–410.
19. Shinya M, Fournier A. Stochastic motion: motion under the influences of wind. *Computer Graphics Forum* (*Proceedings of Eurographics '92*) 1992; **11**(3): C-119–128.
20. Stam J, Fiume E. Turbulent wind fields for gaseous phenomena. In *Proceedings of SIGGRAPH '93*, 1993; 369–376.
21. Peitgen H, Saupe D. *The Science of Fractal Images*. Springer: Berlin, 1988.
22. Lugt H. *Vortex Flow in Nature and Technology*. Wiley–Interscience: Chichester, 1983.
23. Roads C. *The Computer Music Tutorial*. MIT Press: Cambridge, MA, 1996.
24. Computer Graphics Laboratory at Iwate University. http://www-cg.cis.iwate-u.ac.jp/lab/gallery7-e.html [April 2004].

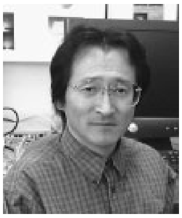
Authors' biographies:

Katsutsugu Matsuyama is currently a Ph.D. candidate in Computer Science at Iwate University. He is engaged in research on sound generation for computer animation. He received the B.E. and M.E. degrees in Computer Science from Iwate University in 1999 and 2001, respectively.

Tadahiro Fujimoto is currently a Lecturer in the Department of Computer and Information Sciences at Iwate University. His research interests include computer graphics, geometric model, and mathematics for shape description in general. He received a BE in electrical engineering, and an ME and Ph.D. in computer science from Keio University in 1990, 1992, and 2000, respectively. He worked at Mitsubishi Research Institute from 1992 to 1995. He was a Research Associate in the Department of Computer and Information Sciences at Iwate University from 1999 to 2002. He is a member of IEICE Japan, IPS of Japan, IEEE and ACM.

Kazunobu Muraoka is currently a Professor in the Department of Communications Engineering, Tohoku Institute of Technology. His research interests include computer graphics and computer animation. He received a BE in electronics engineering from Tohoku

Institute of Technology in 1976 and a DE in electronics and information science from Iwate University in 1999. He was a research student at Tohoku Institute of Technology from 1976 to 1977, a Lecturer at Morioka Junior College from 1989 to 1991, an Associate Professor from 1991 to 1999, and an Associate Professor at Iwate University from 1999 to 2002. He is a member of IEICE Japan, IPS of Japan, IEEE and ACM.



Norishige Chiba is currently a Professor in the Department of Computer and Information Sciences at Iwate University. His research interests include computer graphics, algorithm theory and science on form. He received a BE in electrical engineering from Iwate University and an ME and DE in information engineering from Tohoku University in 1975, 1981 and 1984, respectively. He worked at Nippon Business Consultant Co., Ltd from 1975 to 1978. He was a Research Associate in the Department of Communication Engineering at Tohoku University from 1984 to 1986, an Associate Professor of Computer Science at Sendai National College of Technology from 1986 to 1987 and an Associate Professor of the Department of Computer and Information Sciences at Iwate University from 1987 to 1991. He is a member of IEICE Japan, IPS of Japan, IEEE and ACM.