

Fourier Descriptor Animation of Cloth Flapping in Wind

Janyl Tynystanova¹⁾ Norishige Chiba²⁾ (Member)

1) Department of Design and Media Technology, Graduate School of Engineering, Iwate University

2) Department of Design and Media Technology, Iwate University

{janyl@cg., nchiba@}cis.iwate-u.ac.jp

Abstract

In this paper we propose an efficient new method for modeling the natural motion of cloth flapping based on a stochastic method for wind simulation and a P-type Fourier descriptor. Many have studied cloth behavior through modeling and animation, but only a few studies have combined a stochastic method for wind generation with the physical properties of cloth motion. However, the integration of artificial forces may alter the flapping and fluttering motion. We do not use physical simulation and forces; instead we apply the generated wind value directly to the cloth motion. We apply the $1/f^\beta$ noise in the angles that define the curvature of fibers, using a Fourier descriptor because of its smoothing properties. This allows us to create a natural cloth shape. Since this approach does not require physical simulation based on equations describing the motion, we can easily tune the implementation using parameters that control the cloth motion in real time. Moreover, we propose a few simple techniques to improve the visual realism of fluttering, the appearance of swirling, and wrinkle effects.

1. Introduction

The modeling and animation of interactive cloth behavior is a difficult challenge in the fields of computer graphics (CG) and virtual reality. The modeling of a flag flapping or a banner fluttering involves complicated dynamics that depend on the surrounding wind. To model wind generation and its effect on fluttering motion we must implement a cross-simulation. One possible model is the well-known mathematical model of the energy spectrum of turbulence, derived by Kolmogorov [1]. The power spectral density of $1/f^\beta$ noise is similar to the wind-speed turbulence spectrum, and hence, this noise can be used for the stochastic approximation of wind.

Recently, researchers [3, 16] have tried to simulate curtain behavior using noise generation with gravitational force for the motion. The results were good, but the combination of noise with global physical properties such as attractive, compressive, and gravitational forces leads to slow and unnatural flag animation.

Our model for cloth motion is based on the generation of $1/f^\beta$ noise. Generally, the visualization of the fluttering behavior of cloth by simulating the interaction with simulated wind field provides physical accuracy and visual reality. In our approach, the statistical wind character is introduced directly into the cloth motion using the Fourier

descriptor for open curves.

This provides a balance between simplicity and the realistic simulation of the interaction between cloth and turbulent blowing wind that occurs in flag flapping, banner fluttering, and curtain waving.

Natural wind is non-frozen, but Taylor's frozen turbulence hypothesis is often employed in applications because of its simplicity. In the case of frozen wind, the static turbulence component translates along the wind direction.

The rest of this paper is structured as follows. In Section 2, we discuss the modeling of cloth motion. In Section 3, we give an overview of the process of $1/f^\beta$ noise generation and present the basis of open-curve formation using a P-type Fourier descriptor. In Section 4, we describe our approach to the modeling of cloth motion. We also present algorithms for motion modeling under the influence of frozen and non-frozen wind based on applying noise in 2D and 3D space, respectively. The animation and performance are discussed in Section 5, and the results are given in Section 6. Section 7 provides concluding remarks.

2. Previous works

A survey of the available methods for the modeling of cloth for structural engineering and computer graphics

applications is given in [2].

Most of the existing techniques are based on physical models. For example, Terzopoulos and Fleischer [6] proposed an approach in which the relevant forces are determined by implicit integration techniques. In 1987, the same authors, in cooperation with Platt and Bar, gave a hybrid formulation for the approximation of large cloth deformations [5].

Later, some researchers, particularly Breen et al. [4], suggested so-called particle-based techniques. Platt and Barr [11] developed a technique of controlled soft-object animation. This technique uses mathematical constraints derived from physics and optimization theory.

Ling et al. [8, 9] developed a technique based on aerodynamic theory for modeling cloth deformations induced by airflow.

Bridson et al. [12] developed an approach for the realistic representation of cloth folds and bend formation. This approach uses a mixed explicit and implicit time integration scheme with a flexing model based on physical calculations.

In this work, we develop a new approach for modeling realistic animation of cloth fluttering under the influence of wind, using Fourier descriptors and a stochastic technique for fractional Brownian motion (fBm) generation. The fBm is also referred to more loosely as $1/f^\beta$ noise. We use a simple model of the wind constructed from the spectral density, which was applied by Voss [15] for the generation of random fields to simulate natural phenomena. This model has been successfully employed in computer graphics for the synthesis of trees and grass under the influence of a wind field [13]. In another investigation [10], a combination of noise and simple dynamic simulation provided a new approach to the animation of the natural movement of tree leaves and branches under the influence of a wind field. Stam [14] obtained impressive results by using stochastic models for natural phenomena, specifically volumetric blobs, to create his models of smoke and clouds. A noise-based approach has also been used to animate cloth and curtain flapping [16].

The goal of this work is to offer a method that is convenient and simple in implementation for creating believable animation of cloth flapping under wind.

The major difference between previous approaches and our method is that we generate cloth motion using a noise-based approach, rather than physical-based simulation. That is, our method doesn't require the knowledge on fluid dynamics, elastic dynamics and their computational simulation methods. It is for this reason that we call our approach "animation", instead of "simulation".

3. P-type Fourier descriptor for waving motion

We modeled externally influenced cloth motion by applying 2D and 3D $1/f^\beta$ noise; the values are represented as angles of curve inflexions in the Fourier descriptor.

Two-dimensional noise is often used for the simulation of natural stochastic phenomena such as wind flows, the movements of branches and of leaves on trees, waterfall noise, and smoke diffusion [15, 13, 14, 10]. We used this kind of noise to model wind. Simple and not excessively time-consuming calculations allow us to create flexible models on the basis of the $1/f^\beta$ noise. The models react quickly to any changes in the conditions and parameters made by the animator in real-time. The use of 2D noise means that we cannot implement certain peculiarities of cloth motion, for example, the formation of fine non-recurrent wrinkles, folds, gentle fluctuations, and swirling. However, it is possible to animate the effect of wrinkles even on the basis of frozen wind using a noise-shifting algorithm. The changing of wind speed is achieved by sequential shifting of values in the current row of the 2D noise matrix in the direction of the wind.

For the animation of cloth motion under the influence of non-frozen wind, we use a noise of dimension an order higher than the dimension of the considered object. We generate the speed of noise transpositions in the $|\vec{v}(t)|$ -horizontal and the $|\vec{u}(t)|$ -vertical directions.

3.1 $1/f^\beta$ Noise model

$1/f^\beta$ noise can be represented as signals with spectral density proportional to $1/f^\beta$ [15], where β is the correlation of the noise as it varies over time. The spectral density for the three-dimensional case $S(f_1, f_2, f_3)$ can be calculated as

$$S(f_1, f_2, f_3) = \frac{C \cdot Nrand}{\sqrt{f_1^2 + f_2^2 + f_3^2}^{(\beta+2)/2}}, \quad (1)$$

where $C \cdot Nrand$ assigns the calibration factor, C is constant, and $Nrand$ is a normal random number ranging over $[0, 1]$. For our animation, the noise is visually similar, if $Nrand$ is chosen as constant. Hereafter, because of the simplicity of equation (1), the numerator is denoted as C . In our implementation, C is set as 20,000. The fBm always corresponds to a characteristic frequency spectrum $1/f^\beta$, where β defines the roughness of noise. The peculiarity of this noise is that the power spectrum of $1/f^\beta$ noise is inversely proportional to the degree β of frequency f . Thus, if β increases, the noise decreases, and vice versa. This

property makes the noise controllable. For simplicity, we used an algorithm based on fast Fourier transforms (FFT) to generate the noise.

The generation of one-dimensional $1/f^\beta$ noise based on spectral density is achieved in two steps:

Step 1. Define the frequency spectrum $S(k)$ and the Fourier transform $F(k)$, $0 < k < N/2$ as follows:

$$S(k) = \frac{C}{k^2},$$

$F(k) = S(k) \cos(2\pi \cdot \text{rand}) + iS(k) \sin(2\pi \cdot \text{rand})$, where i is an imaginary number $\sqrt{-1}$, and $F(k)$ and $F(N-k)$ are assumed to be conjugate, i.e.,

$$F(k) = F^*(N - k).$$

Note that the direct-current component $F(0)$ takes an arbitrary real number, because it influences only the average noise value, while $F(N/2) = 0$.

Step 2. Calculate the inverse Fourier transform (IFT) $f(n)$ of $F(k)$ using inverse fast Fourier transform (IFFT). As in the above algorithm, the two-dimensional noise can be defined in terms of the following steps:

Step 1. Define the frequency spectrum $S(k, l)$ and the Fourier transform $F(k, l)$, $0 < k, l < N/2$.

$$S(k, l) = \frac{C}{\sqrt{k^2 + l^2}^{(\beta+1)/2}}$$

$F(k, l) = S(k, l) (\cos(2\pi \cdot \text{rand}) + i \sin(2\pi \cdot \text{rand}))$.

Note that $(\beta + 1)$ should be $(\beta + n - 1)$ for n -dimensional noise. The conjugate is assigned as follows:

$$F(k, l) = F^*(N - k, N - l), \quad k, l > 0$$

$$F(k, N - l) = F^*(N - k, l), \quad k, l > 0$$

$$F(0, l) = F^*(0, N - l), \quad l > 0$$

$$F(k, 0) = F^*(N - k, 0), \quad k > 0$$

Step 2. Calculate IFT $f(m, n)$, $m, n = \overline{0, N - 1}$.

3.2 P-type Fourier descriptor

Fourier descriptors are widely used for the description of curves on a plane surface in the frequency domain. There are well-known descriptors of the Z-type [18], G-type [19], and P-type [17] (in Japanese). The Z-type and G-type descriptors are more appropriate for closed curves. We used the P-type descriptor, because it accurately reproduces both closed and open flexing curves. The low-frequency spectrum contains significant data on the flexing curve, with the result that in the contour reconstruction, after transformation using filters and, in particular a low-pass filter (LPF), the ends of the initial and reconstructed curves coincide one-to-one. The main idea of this approach to curve description is as follows. Given a curve in the plane, we assume that all the segments composing this curve are of equal length.

$$\delta = |\mathbf{P}(j) - \mathbf{P}(j - 1)|, \quad j = \overline{0, n - 1} \quad (2)$$

The angle formed by vectors $\mathbf{P}(j) - \mathbf{P}(j - 1)$ and $\mathbf{P}(j + 1) - \mathbf{P}(j)$ is denoted as $\alpha(j)$ in Fig. 3.1, where $\alpha(0)$ is the angle between the first vector and the X-axis. Angle $\alpha(j)$ is called an angular bend in the point $\mathbf{P}(j)$. It is usually in the interval $-\pi \leq \alpha(j) < \pi$.

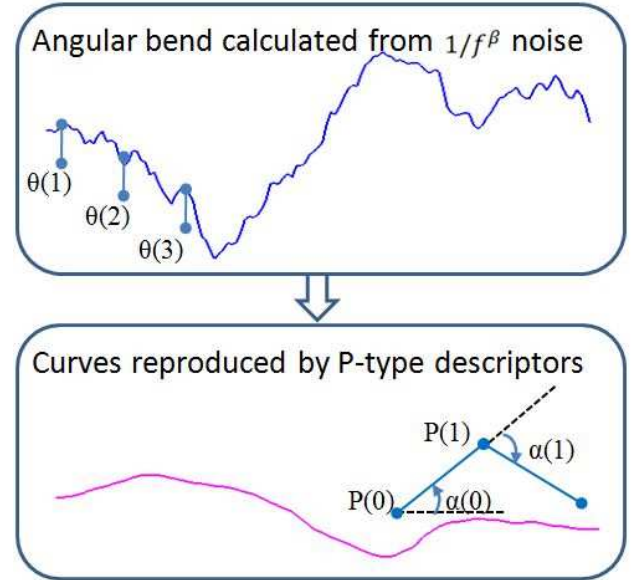


Figure 3.1: Shape description of curves by angles generated as $1/f^\beta$ noise.

A total curvature function of the polygonal line under consideration is defined as

$$\begin{cases} \theta(0) = \alpha(0) \\ \theta(j) = \theta(j - 1) + \alpha(j), \quad j = \overline{0, n - 1}. \end{cases} \quad (3)$$

Next, a function of the complex variable ω is defined by Eq. (4) for a value of θ :

$$\omega(j) = \exp(i \cdot \theta(j)), \quad j = \overline{0, n - 1}. \quad (4)$$

This is a representation of the polygonal line in the form of a P-type descriptor.

Thus, a discrete Fourier transform in relation to the polygonal line considered in terms of the descriptor has the form

$$S(k) = \frac{1}{N} \sum_{j=0}^{n-1} \omega(j) \exp(-2\pi i \frac{jk}{n}) \quad (5)$$

Next, applying an IFT, the approximation of the initial curve has the form

$$\omega(j) = \sum_{k=0}^{n-1} S(k) \exp(2\pi i \frac{jk}{n}) \quad (6)$$

In our case, pre-calculated $1/f^\beta$ noise of the given parameters is applied to the input of the P-type descriptor in the form of angles, $\theta(j)$. We can apply an appropriate filter to $S(k)$ in frequency domain, if necessary.

The integration of random number, fBm of $\beta = 0$, becomes Brownian motion, fBm of $\beta = 2$. Similarly, using noise in the form of angles of a P-type descriptor is equivalent to the calculation of an integral for $\sin(\text{noise})$ that results in a significant smoothing effect.

This property allows us to assume that for modeling node motion in a rectangular grid, we can apply a combined calculation of a P-type Fourier descriptor and $1/f^\beta$ noise, instead of solving the differential equation.

4. Animation of cloth flapping

4.1 Motion in frozen wind

To animate the behavior of flexible objects under the influence of frozen wind, we used a pre-computed matrix of $1/f^\beta$ noise. In several applications, Taylor's frozen turbulence hypothesis is employed for implementation simplicity [20]. The algorithm, presented in Fig. 4.1, has the following steps:

- Step 1. Initialize the noise parameter and the size and scale of the cloth grid.
- Step 2. Pre-calculate a 2D table of noise data.
- Step 3. Rearrange the noise values (see explanation below).
- Step 4. Generate the 2D grid surface using the noise generated in step 2 and re-assembled in step 3, using a P-type Fourier descriptor to calculate the positions of the grid nodes.
- Step 5. Satisfy the constraint (see explanation below).
- Step 6. Perform the animation.

Step 7. Change the conditions interactively at the request of the user. After changing the wind parameter, go to step 2; for other parameters go to step 3.

For step 3, rearrange the noise values of one column by a given step. In flag motion, we assume that wind propagates from left to right, so all noise values of the current column of the 2D noise matrix shift one step further along the cloth grid. The “virtual” velocity of wind is determined from the number of steps of noise value shifting. The bigger the shifting step, the higher is the velocity of the wind. According to the relation between virtual wind velocity and number of shifted steps of noise values in each column, the speed $|\vec{v}(t)|$ in practice is reasonable to adjust within the interval [1, 10]. Using the noise transposition procedure, we can obtain frozen wind.

For step 5, satisfy the constraint on the distance between neighboring grid rows, using the 2D noise data to make a model of the noise movement along the nodes of the rectangular grid that represents the cloth. This model describes the movement from one node to the next, step by step, at every time cycle, thereby providing wave-like bending propagation along the wind direction that is reminiscent of flag flapping. This is shown in Fig. 4.2.

4.2 Motion in non-frozen wind

In practice, the visualization of flag flapping or curtain movement requires a more detailed reconstruction of the physical model. To enhance the visual reality and to provide a better realization of the real motion, we developed an

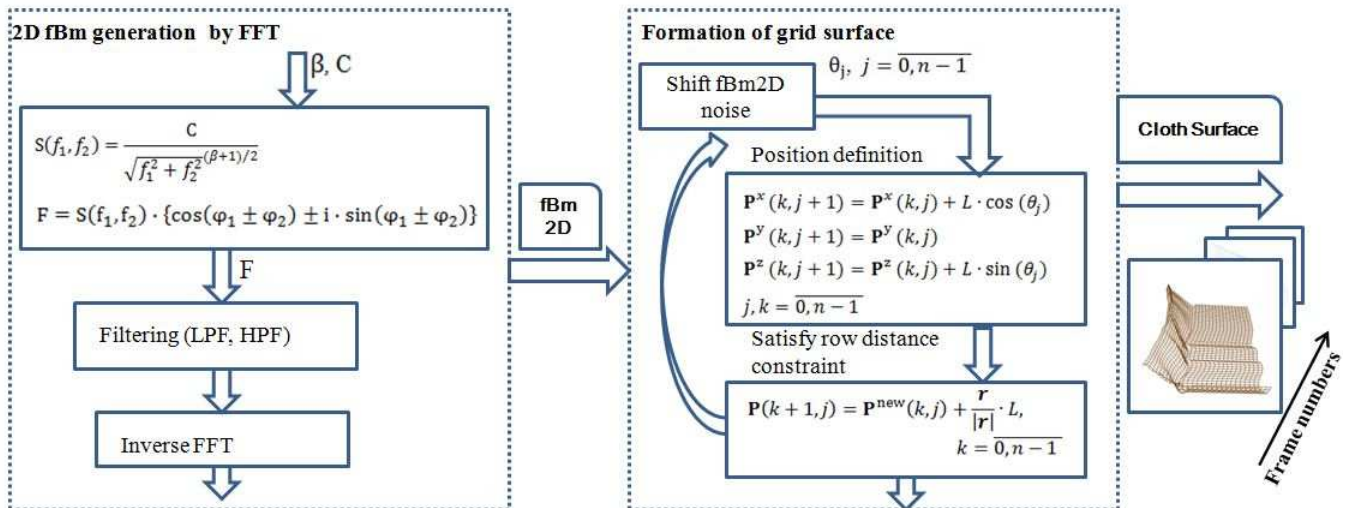


Figure 4.1: Cloth surface generation for each frame under frozen wind, where L is the threshold distance of two neighboring lines, $\mathbf{P}(k, j)$ is the current position of grid node, and $|r|$ is the current distance between two nodes located on neighboring grid rows.

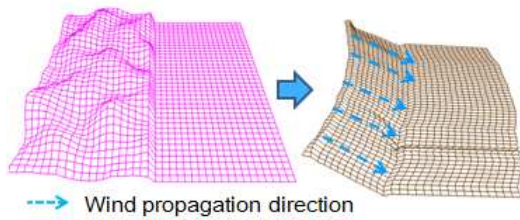


Figure 4.2: Wind propagation on cloth surface.

animation algorithm based on an increase in the order of the noise matrix dimension relative to the modeled object. Approximating wind in this way makes the implementation of complex cloth surface deformation possible, because the wind value changes in 3D space with each time step. We will call this type of wind approximation non-frozen.

This algorithm has the following steps:

- Step 1. This is step is the same as that for frozen wind.
- Step 2. Generate the 3D noise data using the 3D noise matrix according to Eq.(1), where $n = 3$, and the algorithm mentioned in Section 3.1.
- Step 3. Rearrange the noise values of one row by a given step from the end to the beginning (see explanation below).
- Step 4. Apply the noise matrix to the P descriptor (see explanation below).
- Step 5. Generate the animation.
- Step 6. Adjust the noise parameters, and then go to step 2.

For step 3, the noise shifting procedure for non-frozen wind approximation through the example of rope motion is as follows. Each row of 2D noise data represents one frame. Thus, the elements of the first row are not changed; shifts are executed beginning from the second row. For example, in a one-step shift, one element is rearranged from the end of the second row to the beginning. The whole line is shifted to the right by one step. In the third row, two elements at the end are shifted to the beginning in such a way that the next-to-last element becomes the second and the last element becomes the first, and so on. Similarly, for a two-step shift, we rearrange two elements from every row. This procedure is shown in Fig. 4.3.

In the case of 2D object motion, the vertical slices of the 3D noise matrix determine the shape of the cloth for one frame. To transpose a frame from left to right, we perform a shifting procedure in the XZ plane, which defines the horizontal direction of the wind velocity, $|\vec{v}(t)|$. In the XY plane, for the swirling effect, we use the vertical direction of

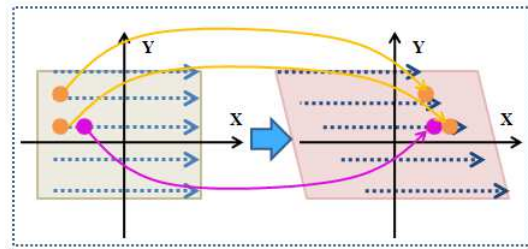


Figure 4.3: Shifting of 2D fBm.

the wind velocity, $|\vec{u}(t)|$. Thus, $(|\vec{v}(t)| + |\vec{u}(t)|)$ is the diagonal direction of the shifting procedure. The scheme for cloth formation in non-frozen wind is shown in Fig. 4.4.

For step 4 apply every line of the noise array to an entry of the P descriptor. The resulting set of lines represents one frame. The bending curve represents one frame by considering the input noise value as a curvature angle. The fluctuation function of $1/f^\beta$ defines the degree of the curvature angle.

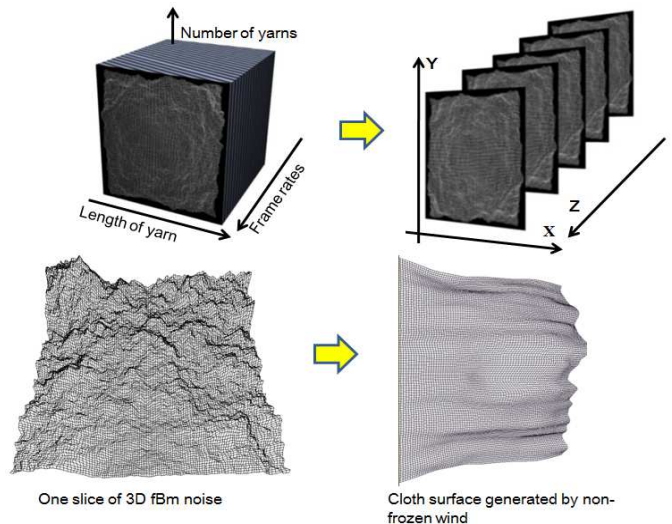


Figure 4.4: Animation technique for cloth flapping motion with non-frozen wind.

5. Animation and performance

5.1 Material and geometrical characteristics of cloth

In our model, a cloth is represented geometrically as connected nodes in a 2D cell grid.

The displacement of noise in the horizontal direction generates a movement, intersections of the horizontal and vertical lines of the grid are called nodes, and the lines connecting these nodes are called horizontal and vertical branches. The sets of horizontal branches are called yarns (Fig. 5.1).

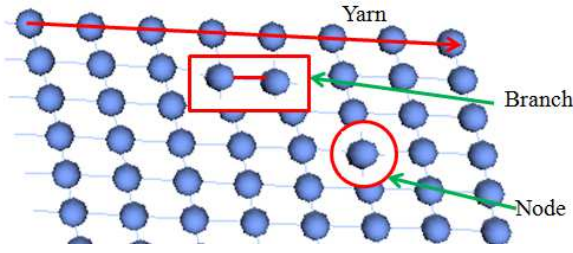


Figure 5.1: Textile structure in the model.

The movements in space are in agreement since they are formed from noise. Specifically, in the case of a 2D noise, the same noise values are transmitted vertically to all nodes of the curtain and every yarn is fixed at one end, but another end moves freely. The vertical distances between the yarns remain uncontrolled. For this reason, a non-dimensional cloth stretching is observed during the motion. However, the length of each thread remains unchanged as a result of the equality of the segments that form one yarn. This peculiarity comes directly from the definition of the Fourier descriptor. Hence, to provide flexibility and elasticity to the cloth, it suffices to maintain definite distances between yarns and, for elasticity, to satisfy the appropriate scale of the cell grid.

One of the most common approaches for obtaining the required cloth properties is the spring model. In the application of this model, it is necessary to recalculate distances between close and distant neighbors many times at every stage. In our model after cloth shape formation, each node position is corrected according to relationship (7). Note that in our model we perform no controlling if the distance between vertical branches is small, because this provides folds on the cloth surface; instead, we ensure that it does not exceed a given threshold distance L . Thus, to maintain a given distance between yarns, the yarns should be aligned relative to the indicated yarn. This yarn is called the leading yarn. The alignment may be done relative to the uppermost or lowest yarns, which correspond to the upper and lower cloth edge, respectively.

$$\mathbf{P}(k+1, j) = \mathbf{P}^{\text{new}}(k, j) + \frac{\mathbf{r}}{|\mathbf{r}|} \cdot L, \quad (7)$$

$$k = 0, n-1$$

where $\mathbf{P}(k+1, j)$ is the position of the node after correction, $\mathbf{P}^{\text{new}}(k, j)$ is the position of adjacent of the node after correction, \mathbf{r} is a vector defined as $\mathbf{P}(k, j) - \mathbf{P}(k+1, j)$, and L is the threshold distance between two vertical nodes.

The elongation of the left edge of a flag is defined by the wind strength assigned by comparing our observations with the Beaufort scale. Thus, the coordinates of the nodes on the left edge are

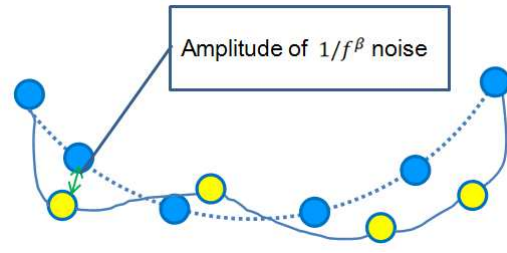


Figure 5.2: Cloth edge modeling.

$$\begin{aligned} \mathbf{P}^x(k, 0) &= W_1 \cos(\gamma + \alpha_k), \quad k = \overline{0, n-1} \\ \mathbf{P}^y(k, 0) &= -W_2 \gamma (1 + \alpha_k), \\ \mathbf{P}^z(k, 0) &= n \cdot k, \quad \gamma = \frac{\pi}{2}, \quad \alpha_k = \frac{k\pi}{n}, \end{aligned} \quad (8)$$

where n is the number of nodes, W_1 is the vertical elongation, and W_2 is the horizontal elongation (see Fig. 5.2).

Note that the properties of cloth, such as branch or yarn length, mean “virtual” size and do not match the real physical size of cloth. Giving a different number of cells in a grid makes it possible to change the visual impression, such as the thickness and size of cloth. Essentially, in our implementation, the scale of a curtain model was chosen to have branch-width = 0.6 and branch-height = 0.4, and that of a flag and banner model was set as 0.5 and 0.3, respectively, but with a different size grid.

5.2 Real-time control of motion

Animators often need a model that is realistic and has easy-to-use tools for parameter tuning. We have achieved a compromise between the accuracy of the model and the time needed to create it. Our model has the following interactive parameters:

1. Wind speed
2. Vibration amplitude
3. Degree of cloth smoothness and winding
4. Swirling value
5. Twisting motion
6. Flattening line

The wind speed for 2D $1/f^\beta$ ranges from 1 to 10. In our case, a wind speed of fresh breeze on the Beaufort scale was visually reproduced by setting $|\vec{\mathbf{v}}(t)| = 10$ and 50 fps. The vibration amplitude α , which defines the angle of deviation relative to the vertical, is in the interval $[-\pi, \pi]$ it defines. The amplitude of cloth fluttering corresponds to wind speed, with greater amplitude, corresponding to higher wind speed. We observed experimentally that for the modeling of flag and banner flapping in gentle breeze, a sufficient range for the peak-to-peak amplitude is $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

Table 1. Observations for wind speed with parameters of our performance.

Beaufort scale (m/s)	Observation	Model	θ	β	$ \vec{v}(t) $	$ \vec{u}(t) $	BPF range
Light Breeze 2.7–3.6	Banner lightly fluttering	frozen	$[-1.4, 1.4]$	2.5–3.0	1	0	[3, 7]
		non-frozen	$[-1.4, 1.4]$	2.5–3.0	2	1	[3, 7]
Gentle Breeze 3.6–7.2	Banner intensively flapping	frozen	$[-1.4, 1.4]$	2.0–2.5	2	1	[3, 7]
		non-frozen	$[-1.4, 1.4]$	2.0–2.5	8	0	[1, 7]
Gentle Breeze 3.6–7.2	Light curtain moving toward wind	frozen	$[-1.9, 1.9]$	2.0–3.0	2	1	-
		non-frozen	$[-1.9, 1.9]$	2.5–3.0	4	0	-
Moderate Breeze 8.9–12.5	Curtain intensively fluttering	frozen	$[-2.8, 2.8]$	2.0–3.0	1	0	-
		non-frozen	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	2.5–3.0	10	1	[1, 6]
Gentle Breeze 3.6–7.2	Light flag flapping	frozen	$[-1.4, 1.4]$	2.5–3.0	2	1	[3, 5]
		non-frozen	$[-0.8, 0.8]$	2.0–3.0	2	1	[3, 6]
Moderate Breeze 8.9–12.5	Uncurling of flag	frozen	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	2.8–3.0	3	1	[3, 5]
		non-frozen	$[-2.0, 2.0]$	2.7–3.0	4	0	[3, 6]

The degree of cloth smoothness may be given by a combination of the values for β and θ .

Given the different number of steps (1–4, 8, and 10) for shifted noise values, we can change the virtual wind velocity $|\vec{v}(t)|$ of wind. In this way, we learned experimentally that $|\vec{v}(t)| = 1, 2$ corresponds to a light breeze with speed 2.7 – 3.6 m/s, whereas choosing $|\vec{v}(t)| = 4, 8$, models a wind with a 6 – 7.2 m/s speed. On the other hand, by giving a step of noise transposition in the vertical direction \vec{u} we can obtain a swirling effect. In our implementation, only 0, 1 and 2 values occurred (see Table 1).

We can also choose a flattening line interactively. There are three choices: the alignment can be done along the upper, central, or lower edge of the grid.

The computational times for 2D and 3D $1/f^\beta$ noises are 0.219 s and 2.798 s, respectively, where size for each 2D matrix is 256×256 .

6. Results

In this section, we present the results of cloth flapping animation. The wind velocity in our animation is adjusted intuitively. Thus, wind velocity for flag and banner animation corresponds to a number of shifted elements of the noise matrix in the horizontal direction \vec{v} ; on the other hand, the magnitude of folds is determined by the number of shifted elements in the vertical direction \vec{u} . In order to make the cloth flapping animation under the approximated wind believable, it was compared with a description of real objects (curtain, flag, and banner) in accordance with the Beaufort scale.

We confirmed that the motion was visually realistic in comparison with real motion in a video of these objects. Figs. 6.1 and 6.2 illustrate comparative results of video capture with animation. The animation results were obtained using a computer with an Intel Core (2.0 GHz) processor, 2 GB RAM, and an NVIDIA GeForce 9400GT video card.

6.1 Fluttering motion

The results of the curtain fluttering motion based on 2D and 3D noise are shown in Figs. 6.3 and 6.4. With a left leading yarn and a swirling value for a frozen wind, we observed wind flow from the left side. This performance suggests a slow wind, described in Table 1 as a gentle breeze.

6.2 Flag and banner in frozen/non-frozen wind

Flag flapping under a strong wind (a moderate breeze) was performed using a band pass filter (BPF), which allows wrinkle motion (see Fig. 6.5). Fig. 6.6 shows the results of flag animation for a frozen wind (gentle breeze). Fig. 6.7 shows a screenshot of animated results of a banner for frozen/non-frozen wind. The figure shows results of these tests with different wind speeds, as well as changes in other parameters.

The results for cloth motion modeling confirm the usability of our method based on the combined application of $1/f^\beta$ noise and a Fourier descriptor.

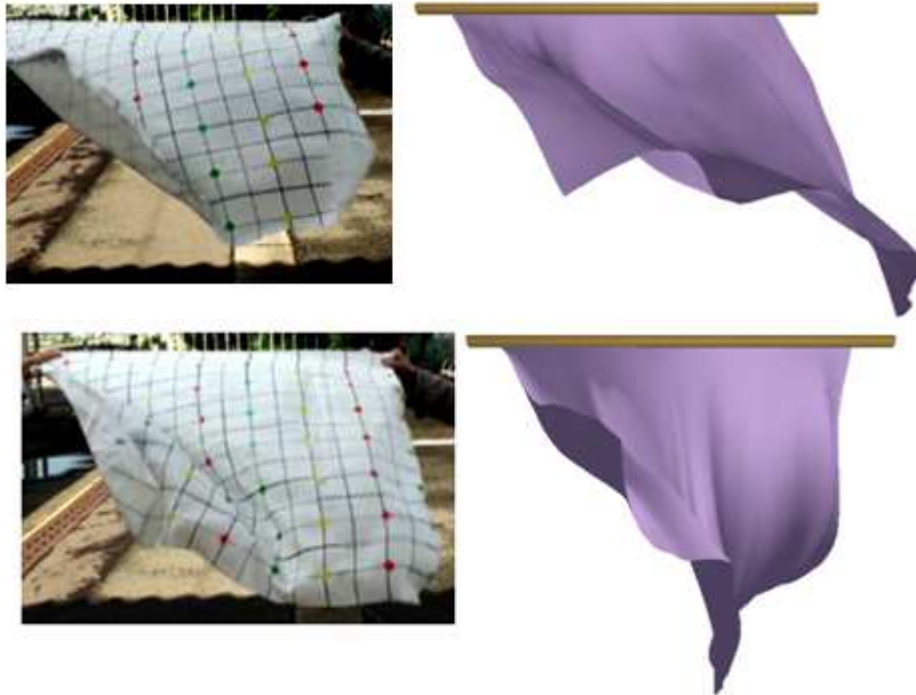


Figure 6.1: The left picture is a video capture of cloth flapping in gentle breeze; the right is a screenshot from cloth animation in a non-frozen wind with the following parameters: $\beta = 2.7$, $|\vec{v}(t)| = 2$, $|\vec{u}(t)| = 1$, $\theta = [-1.9, 1.9]$.



Figure 6.2: Comparison between video capture and animation results of flag flapping in a frozen wind (light breeze): $\beta = 2.7$, $|\vec{v}(t)| = 2$, $|\vec{u}(t)| = 1$, $\theta = [-1.4, 1.4]$.



Figure 6.3: Light curtain animation in a frozen wind: $\beta = 2.7$, $|\vec{v}(t)| = 2$, $|\vec{u}(t)| = 1$, $\theta = [-1.9, 1.9]$.

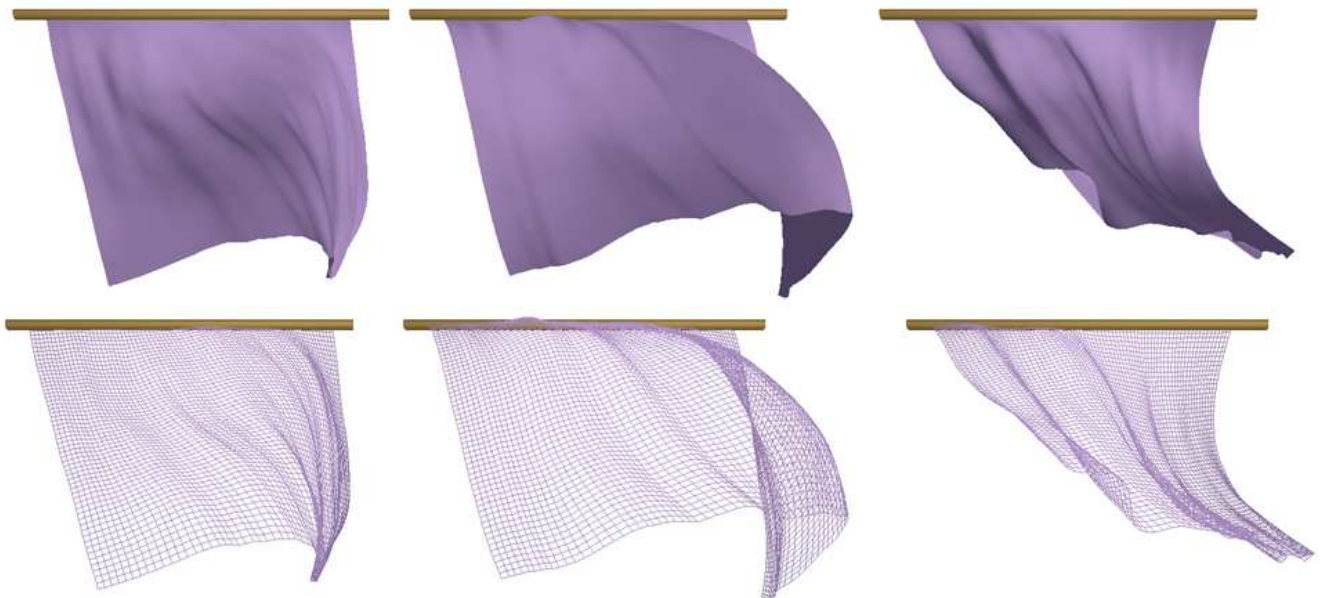


Figure 6.4: Light curtain animation in a non-frozen wind: $\beta = 2.7$, $|\vec{v}(t)| = 4$, $|\vec{u}(t)| = 0$, $\theta = [-1.9, 1.9]$.

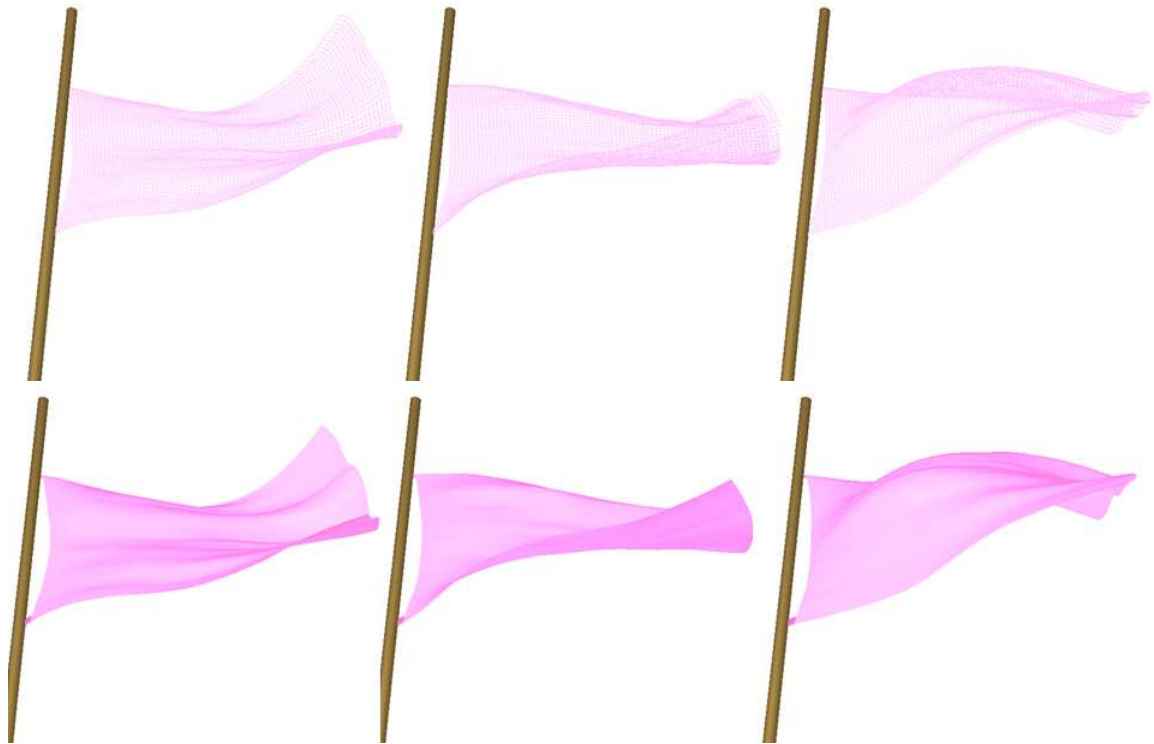


Figure 6.5: Flag animation in a non-frozen wind: $\beta = 2.7$, $|\vec{v}(t)|=4$, $|\vec{u}(t)| = 0$, $\theta = [-2.0, 2.0]$, BPF [3, 6].

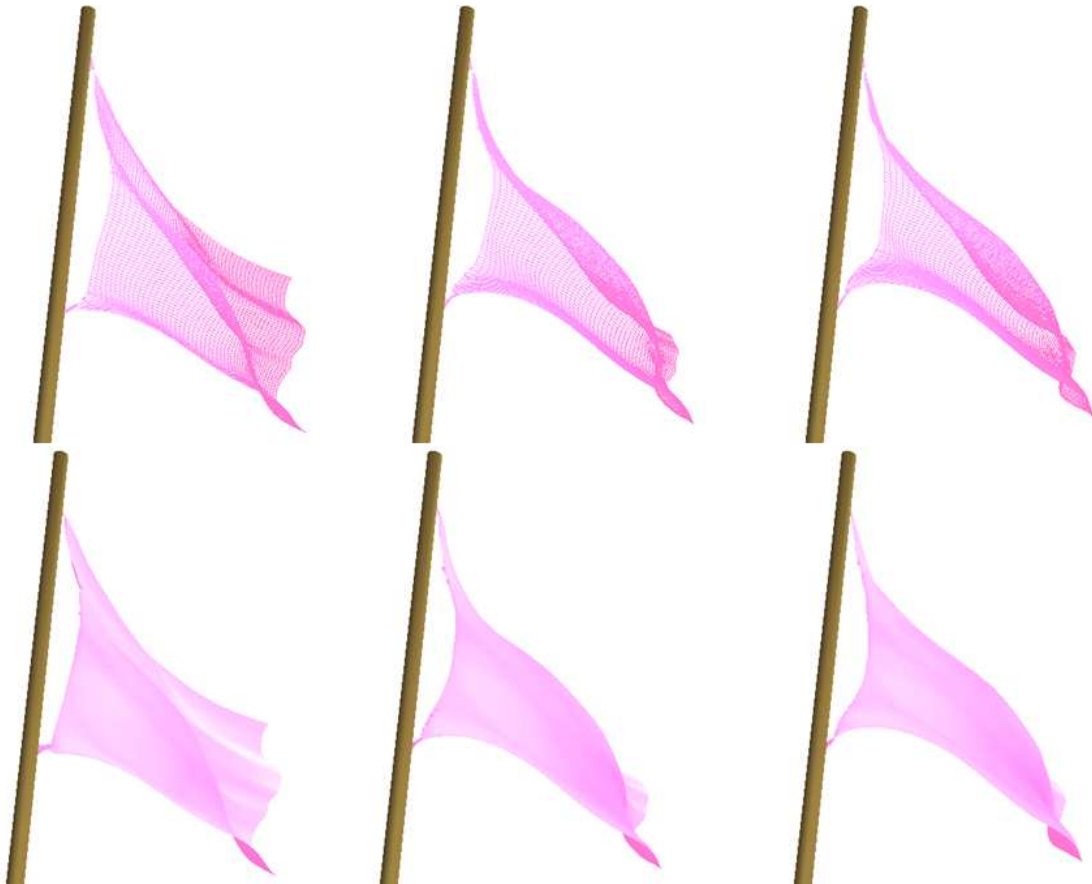
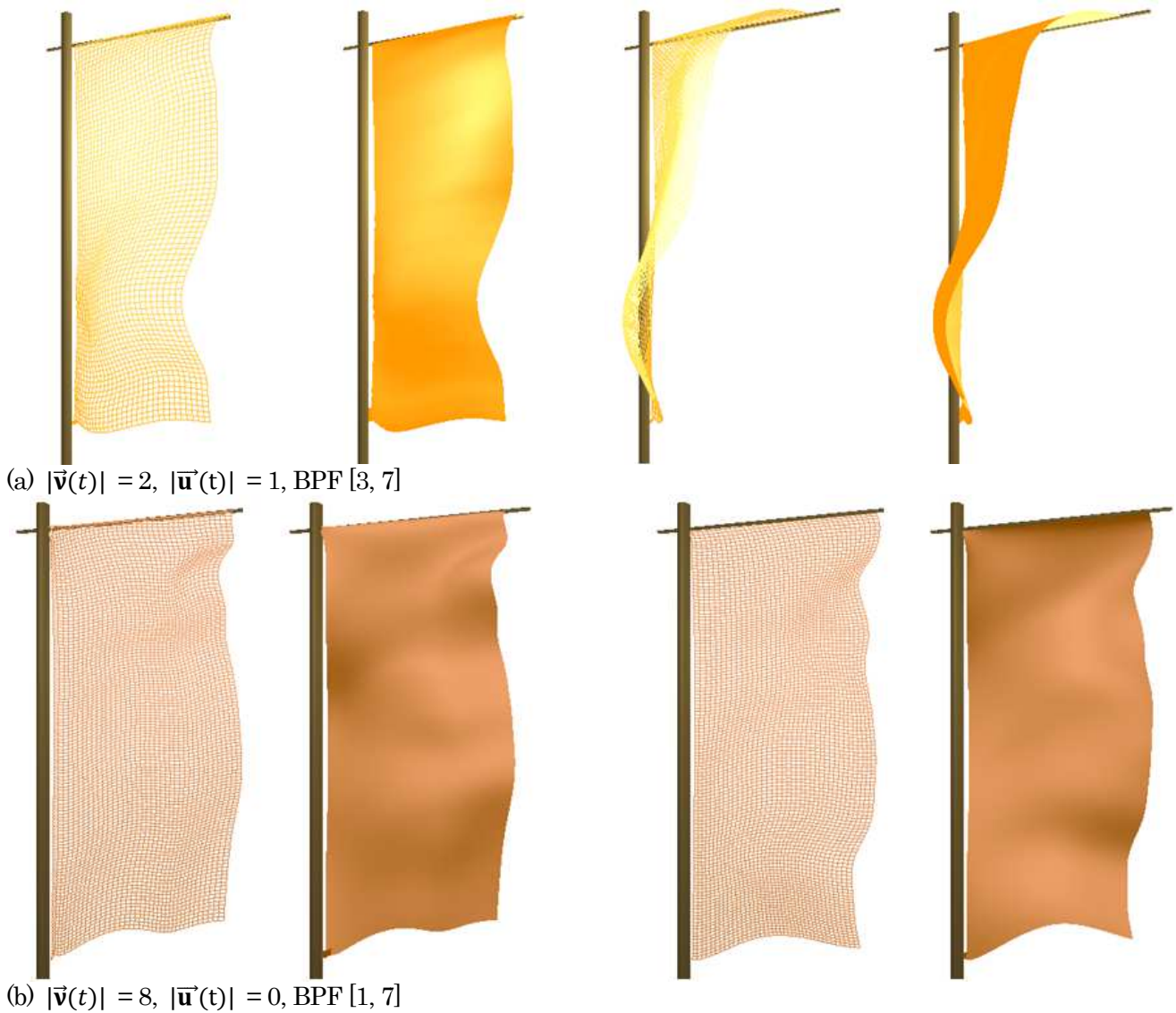


Figure 6.6: Small flag animation in a frozen wind:
 $\beta = 2.8$, $|\vec{v}(t)| = 2$, $|\vec{u}(t)| = 1$, $\theta = [-1.4, 1.4]$, BPF [3,5].



(a) $|\vec{v}(t)| = 2, |\vec{u}(t)| = 1, \text{BPF} [3, 7]$
 (b) $|\vec{v}(t)| = 8, |\vec{u}(t)| = 0, \text{BPF} [1, 7]$
 Figure 6.7: Banner simulation for a) frozen, and b) non-frozen winds: $\beta = 2.5, \theta = [-1.4, 1.4]$.

The model describes flapping and fluttering cloth movements with sufficient accuracy.

In practice, it is important that non-linear and anisotropic cloth motions can be obtained without time-consuming calculations.

7. Conclusions

We have proposed an effective approach for the generation of cloth motion under the influence of wind, with the parameters controllable in real-time. This approach is based on the application of $1/f^\beta$ noise for the wind approximation and uses a P-type Fourier descriptor. We animate the flapping of a flag fixed at two ends with its left edge stretching under the influence of winds of different strengths. We use simple arc-like stretching and determine the far left positions using noise. By applying shifts

in the $1/f^\beta$ noise matrix, we were able to implement twisting and wrinkle effects. The alignment of the grid's longitudinal lines allows us to control the formation of slight bends at the cloth edges. Using the frequency-domain filtering spectrum, the user can operate the formation folds. The user can directly control parameters such as the wind speed or wrinkle effect to visualize the peak-to-peak amplitude of the fluctuations or to simulate the motion of light or heavy cloth.

We have also proposed an animation algorithm that simulates cloth motion under the influence of a non-frozen wind.

Implementing cloth-flapping behavior using our approach is simple; on the other hand, because our method doesn't consider any collision and gravity, our method has no abilities for simulating

interaction with the cloth itself and between the cloth and any other objects.

In future work, we will consider vertical folds on the cloth surface. These often appear in flag flapping.

Acknowledgements

This work was partially supported by KAKENHI(24300032), a Grant-in-Aid for Scientific Research (B). The authors would like to thank Drs SOSORBARAM Batjargal and GUNJEE Zorig for valuable discussion.

References

- [1] A.N. Kolmogorov. C. R. Acad. Sci. USSR 30: 301, 1941.
- [2] M. Aono. A wrinkle propagation model for cloth. Proceedings of Computer Graphics International, 95-115, 1991.
- [3] B. Sosobaram, Z. Gunjee, T. Fujimoto and N. Chiba. Noise-based animation of flag-like objects in a wind field. NICOGRAPH Spring 2010.
- [4] D.E. Breen, D.H. House, and P.H. Getto. A particle-based computational model of cloth draping behavior. Proceedings of Computer Graphics International 113-133, 1991.
- [5] D.D. Terzopoulos, J.C. Platt, A.H. Barr, and K.Fleischer. Elastically deformable models. Computer Graphics (Proc. SIGGRAPH), 21:205-214,1987.
- [6] D. Terzopoulos and K. Fleischer. Deformable models. The Visual Computer, 4:306-331, 1988.
- [7] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In Computer Graphics (Proc. SIGGRAPH), 22, 269-278, 1988.
- [8] Li Ling, M. Damodaran, and Robert K.L. Gay. A model for animating the motion of cloth. Comput. & Graphics, 20, (1): 137-156, 1996.
- [9] Li Ling, M. Damodaran, and Robert K.L. Gay. Aerodynamic force models for animating cloth motion in air flow. The Visual Computer (Proc. Springer-Verlag)12:84-104, 1996.
- [10] S. Ota, M. Tamura, T. Fujimoto, K. Muraoka, and N. Chiba. A hybrid method for real-time animation of trees swaying in wind field. The Visual Computer, 20, 613 -623, 2004.
- [11] J.C. Platt and A.C. Barr. Constraint methods for flexible model. Proceedings of SIGGRAPH Computer Graphics, 22:279-288, 1988.
- [12] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. EUROGRAPHICS/SIGGRAPH Symposium on Computer Animation, 2003.
- [13] M. Shinya and A. Fournier. Stochastic motion: Motion under the influence of wind. Computer Graphics Forum, 119-128, 1992.
- [14] J. Stam. Stochastic dynamics. Simulating the effects of turbulence on flexible structures. Computer Graphics Forum, 159-164, 1997.
- [15] R.F. Voss. Fractals in nature: from characterization to simulation. In The Science of Fractal Images, Springer-Verlag, New York, 21-70, 1988.
- [16] Z. Gunjee, B. Sosobaram, T. Fujimoto, and N. Chiba. Noise-driven approach for animating dynamic natural Scenes, 26th NICOGRAPH 2010.
- [17] Y. Uesaka. A New Fourier descriptor applicable to open curves. IEICE, J67-A, (3): 166-173, 1984.
- [18] C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. IEEE Trans. Computers, C-21, 269-281, 1972.
- [19] G.H. Granlund. Fourier preprocessing for hand print character recognition. IEEE Trans. Computers, C-21, 195-201, 197.
- [20] A. Antonia, A.J. Chambers, and N. Phan-Thien. Taylor's hypothesis and spectra of velocity and temperature derivatives in a turbulent shear flow, Boundary-Layer Meteorology, U.S.A. 19(1): 19-29, 1980.



Janyl Tynystanova received her bachelor's and master's degrees in applied mathematics in Tomsk State University, Russia in 1997. She also received a PhD degree in technical sciences from National Academy of Sciences of the Kyrgyz Republic in 2005 and MS degree in

computer and information science from Iwate University in 2009, respectively she is currently a PhD candidate in computer science at Iwate University. Her research interests include noise-based simulation and animation of natural phenomena in computer graphics.



Norishige Chiba is currently a Professor in the Department of Computer Science at Iwate University. His research interests include Computer Graphics, Laser Graphics and Interactive Graphics. 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 the Department of Computer Science at Sendai National College of Technology from 1986 to 1987 and an associate professor of the Department of Computer Science at Iwate University from 1987 to 1991. He is a member of The Society for Art and Science, IPS, IEEE and ACM.