ABSTRACT: Natural landscape is familiar to our life, and realistic expression by computer graphics (CG) is requested often. Trees are indispensable components in a natural landscape, and their rendering is especially important. In animation, trees sway by wind. Structure of a tree is complex, and even the modeling of static tree is not easy. If the mechanism of tree swaying is simulated based on physical rules, the knowledge of fluid dynamics and material mechanics are required, and huge amount of computation must be executed. In this paper, we extract information on swaying from the video of real trees, and apply it to the CG models of trees. We use ordinary video camera for the video capturing because special equipment such as motion capturing device is expensive and it is difficult to use them in the outdoor environment. We also propose techniques for the video analysis of swaying leaves and branches based on the neighbor image block matching.

Keywords: tree, sway, wind, image processing, Fourier transform, L*a*b* color space

1. INTRODUCTION
Natural landscape is very familiar to us, and its realistic rendering is often requested. Among various components, trees are almost always necessary for the natural scenes, and their realistic expression is very important.

The structure of a tree is complex. Even a still tree is not modeled easily. Trees sway by wind. Accurate physical simulation of swaying tree needs very expensive computation of liquid and material dynamics. Although there are various researches for the rendering of still trees, the research for swaying trees is rare. The goal of our research is to extract information from swaying trees by video analysis, and to apply the information to the CG trees. By extracting the sway information from the real trees, it is expected that the swaying trees can be modeled effectively.

2. RELATED WORKS
Physical simulation is the most important approach for the treatment of trees dynamically. [1] proposes the modeling of trees taking stiffness and weight of them in consideration. In this research, a tree is constructed as a spring-mass model. Each mass has its weight, the distance between two masses does not change, and the stress is considered as a force to resist the bending. Forces are applied to this model, and the form of the tree is determined by the kinetic computation at each mass. This approach needs a lot of computation, and the accurate information of forces to be applied must be supplied. The force generated by wind is difficult to analyze. Therefore it is difficult to express swaying by wind using this model.

[2] proposes a modeling method of the branch structure for an interactive environment. This technique is useful for an animation, too.

In [3], sway by wind is approximated using $1/f^\beta$ noise for real-time animation. A tree is modeled as a collection of connected branches, and the rotation of branches is considered at each node independently. The angle of rotation is given as a $1/f^\beta$ noise for each axis. Their animation gives realistic impression without expensive computation of fluid and elasticity dynamics, but the generated motion does not necessarily reflect the real sway.

3. APPROACH
Difficulty of the tree animation is summarized as follows:

A tree has a hard trunk, a lot of elastic branches, and a lot of soft leaves. The physical characteristics of components vary largely. The structure is very complex, and its modeling is difficult.

A tree sways by wind. The analysis of wind forces applied to each node is very difficult. Especially, the analysis of internal branches and leaves is almost impossible.

An elastic object such as a tree is often
expressed as a mass-spring model. It is a model that connects masses by springs. Each mass is given its weight. The shape of the model is determined by computing the elasticity of the springs, but in our case, the analysis of the forces applied to each part is difficult.

As a tree consists of many parts, the internal parts cannot be seen from the exterior of a tree. Moreover, in natural environments, similar trees often exist in the background. This makes the image and video analysis of a tree very difficult.

A tree has various components, and each part sways differently. A trunk and a thick branch have clear contours and sway largely and slowly. A collection of leaves is recognized as a texture and sways rapidly. Such characteristics must be considered in the video analysis.

We propose the following procedure by considering above points.

[1] Taking video of a tree
[2] Acquisition of motion by video analysis
  (a) Conversion of color model
  (b) Frequency decomposition of lightness image
  (c) Matching of neighbor blocks
  (d) Tracking based on color component
[4] Application of sway information to the CG tree

4. ALGORITHM

4.1 Conversion of Color Model

Ordinary tree exists outdoors, and the use of special equipment is difficult in such environment. We use an ordinary video camera and take a video from one viewpoint. A typical frame is shown in Fig. 1.

As we do not restrict the taking condition, RGB information is affected largely by the environment.

In RGB model, it is difficult to identify one tree from other objects that have similar colors. We convert the RGB color model to L*a*b* color model for the separation of the color information.

4.2 Frequency Decomposition

Each part of a tree sways differently, and we use these characteristics for the video analysis. The information of contour and shape is contained in the lightness component. Therefore, we convert the lightness image to a frequency image by 2D Fourier transformation. The lightness image for Fig. 1 is shown in Fig. 2, and the spectrum image generated from the frequency image is shown in Fig. 3. The size of the original image is 720 x 480 pixels, and the size of the spectrum image is 1024 x 1024 pixels. The spectrum image is symmetric, and the inner position corresponds to high frequencies.

By applying inverse Fourier transformation for the specific frequency range extracted from the frequency image, an image decomposed by frequency is obtained. To reduce the noise of transformation, we apply a Gaussian filter to the

Fig. 1 An example frame of the video.

Fig. 2 A lightness image

Fig. 3 A spectrum image obtained from Fig. 2.
frequency image before the frequency decomposition. An image obtained by this processing is shown in Fig. 4. This image is obtained from the frame in Fig. 1, and the size of the frequency image (Fig. 4(a)) is 1024 x 1024 pixels. In Fig. 4(b), the width of the Gaussian filter is 100 pixels, and the central frequency of the width is varied.

![Frequency Image](image)

(a) Filtering of frequency image.

![Inverse Fourier Transformation](image)

(b) An image obtained by inverse Fourier transformation.

Fig. 4 Frequency decomposition by a Gaussian filter with fixed width.

4.3 Neighbor Block Matching
For each pixel in the frequency images obtained by the frequency decomposition, we search for the corresponding pixel by block matching. In the usual block matching, to find the destination of the area R centered at current pixel, an area with the smallest sum of squared difference $D(x, y)$ is searched in the next frame. This process is shown in Fig. 5.

![Block Matching](image)

Fig. 5 The block matching.

In the video analysis, a lot of block matching must be executed. Branches move continuously, and the amount of movement is restricted in the consecutive frames. Therefore we search for the matched block only in the limited area of the next frame. We adopt the block with the smallest $D(x, y)$ in this search area. This process is shown in Fig. 6.

![Neighbor Block Matching](image)

Fig. 6 The neighbor block matching.

4.4 Tracking based on Color Component
We obtain the optical flow by the neighbor block matching, and we can trace the motion of each part of the tree. In this step, we average the move vectors in small area for avoiding matching errors. To decide the optical flow $F_{t+1}(x, y)$ of pixel $(x, y)$ at frame $t$, we compute the weight $w(x_0, y_0)$ of neighbor pixel $(x_0, y_0)$ using color information. The optical flow vector is decided as the weighted sum of the move vectors $V$ of the neighbor pixels as shown in (1).

$$F_{t+1}(x, y) = \frac{1}{n} \sum_{(i, j) \in R} w(x + i, y + j)V_{t+1}(x + i, y + j)$$

Here, $n$ is the number of pixels in area $R$.

The weight $w(x, y)$ in (1) follows a normal distribution. Suppose that the lightness is $L^*(x, y)$, chroma is $C^*(x, y)$ and hue is $H^*(x, y)$ at pixel $(x, y)$. The distribution of these three components varies by the kind of trees. From the color distribution, we decide the mean values $\mu_L$, $\mu_C$ and $\mu_H$, and variances $\sigma_L^2$, $\sigma_C^2$ and $\sigma_H^2$ for specific tree, and decide the weight of the three components by the following.

$$w_{L^*}(x, y) = \sqrt{2\pi\sigma_L N} \mu_{L^*}, \sigma_{L^*} (L^*(x, y))$$
\[ w_{C*}(x, y) = \sqrt{2\pi}\sigma_C N_{\mu_{C*}, \sigma_{C*}^2}(C^*(x, y)) \]  
\[ w_{H*}(x, y) = \sqrt{2\pi}\sigma_H N_{\mu_{H*}, \sigma_{H*}^2}(H^*(x, y)) \]

Here, \( N_{\mu, \sigma^2}(t) \) means the Gaussian probability density function with mean \( \mu \) and variance \( \sigma^2 \). Finally, the weight of a pixel is determined by (5).

\[ w(x, y) = w_{L*}(x, y)w_{C*}(x, y)w_{H*}(x, y) \]  

Using (1) and (5), the optical flow of each pixel in each frame is determined. This process is shown in Fig. 7.

\[ \Delta l_i = \|P_{i+1} - P_i\| \sin \Delta \theta_i. \]  
\[ \Delta l = \sum_{i=1}^{n} \Delta l_i \]  

Fig. 8 shows the motion tracking data obtained by our tracking process. Fig. 8(a) is the original image, and Fig. 8(b) through 8(d) are the obtained tracking data separated by frequency.

**4.5 Modeling of Trees**

A tree is modeled using a tree structure of node and edges. Parameters considered at a node are shown in Fig. 9.

Fig. 7 The algorithm for tracking.

Fig. 8 Obtained tracking data for various frequency range.

**4.6 Application of Sway to Trees**

We want to move the tip \( P_{i+1} \) of each branch of a CG tree. By the rotation \( \Delta \theta_i \) of the \( i \)th node, we suppose that the tip moves \( \Delta l_i \). As the time interval between two successive frames is small, the relation of \( \Delta l_i \) and \( \Delta \theta_i \) can be approximated by the following.

\[ \Delta l_i = \|P_{i+1} - P_i\| \sin \Delta \theta_i. \]  

The move distance \( \Delta l \) of the tip is expressed as

\[ \Delta l = \sum_{i=1}^{n} \Delta l_i \]
Fig. 9 The model of a node of a tree.

The model for three joints is shown in Fig. 10, and the movement of this structure is shown in Fig. 11.

When the tip of an elastic structure such as a tree is moved, each node translates its movement to the next node, and the energy of the movement is distributed. The branches connected to each other have different thickness in a tree. A thick branch is difficult to move. Therefore, we distribute the displacement of the tip inversely proportional to the thickness of each branch.

\[
\sin \Delta \theta_1 : \sin \Delta \theta_2 : \sin \Delta \theta_3 = \frac{1}{l_{14}} : \frac{1}{l_{24}} : \frac{1}{l_{34}} \]  

(8)

From this consideration, the rotation angle of each node is determined as

\[
\Delta \theta_i = \arcsin \left( \Delta l \left( \frac{1}{l_{i(n+1)}} \right) \sum_{j=1}^{n} \frac{1}{l_{j(n+1)}} \right) \]  

(9)

We can extend this two-dimensional argument to three-dimensional sway by analyzing the movement of the tips, decomposing into horizontal and vertical components, and applying the composite rotation to each node.

4.7 Applications of Sway of Different Frequencies

The sway information is obtained from the images decomposed by the frequencies. The sway extracted from low frequency images can be considered as the sway of thick branches; sway from the high frequency image is considered as the rapid move of small leaves and flowers. By applying this sway to appropriate part of a tree, we obtain tree animation with natural sway.

5. RESULTS

Some frames of the generated animation are shown in Fig. 12. The frame rate is 10 fps, and POV-Ray is used for the rendering.

6. CONCLUSIONS AND DISCUSSIONS

Many parameters are needed for the control of tree motion. In this paper, a method to create animation of swaying trees efficiently without
parameter specification is proposed. That enables the expression of tree motion with lower cost than the past methods. Our work is unique and there seems little work for the acquisition of tree motion using image processing technique.

In our current implementation, the block matching process needs 90 seconds for each frame, and 140 seconds for the tracking process of each pixel. This computation cost must be reduced. We do not check the collision of the branches in the application of sway information. For more complicated tree model the collision check would be required. The editing of obtained motions is an interesting research topic.

We also want to improve the quality of rendering.

ACKNOWLEDGMENTS

This work is supported in part by a Grant in Aid for the 21st Century Center of Excellence from the Ministry of Education, Culture, Sport, Science and Technology in Japan.

REFERENCES