Tree Animation based on Hierarchical Shape Matching

Sang-Min Song, Young-Min Kang, Eung-Joo Lee, and Soo-Yol Ok†

TONGMYONG University
535, Yongdang-Dong, Nam-Gu, Busan, Korea, 608-711
{songsmir,ymkang,ejlee,sooyol}@tu.ac.kr

Abstract. In this paper, we propose a method that produces the realistic motion of L-system based virtual trees. Virtual trees generated with traditional L-system were usually animated with physically-based approaches which require complex motion equations to be solved. Therefore, realtime and plausible animation of such models has been regarded a very difficult problem, and has been rarely employed in interactive applications. The method proposed in this paper plausibly animates the virtual trees based on hierarchical shape matching (HSM) without any complex dynamic equations. The HSM is an improved version of usual shape matching. In this approach, the object is divided into small clusters, and each cluster is locally animated with the shape matching while the global shape is also adjusted in accordance with the original shape.

Key words: Shape matching, virtual tree, L-system, hierarchical animation

1 Introduction

[1] In graphics literature, the tree modeling approaches based on L-system can be easily found[1]. However, those methods do not usually generate the motion of trees because the realistic animation of the complex virtual tree is not a simple problem. The animation of such model usually requires heavy computation for stably solving the motion equation.

In this paper, we propose a hierarchical shape matching(HSM) method. The method is an improved version of usual shape matching. In this approach, an object is divided into small clusters, and each cluster is locally animated with the shape matching while the global shape is also adjusted in accordance with the original shape. The proposed method was successfully employed in a realtime game environment.

2 Previous work

In computer graphics literature, various modeling methods have been intensively studied for realistic representation of the appearance of tree [2–5]. However, those methods focus only on the geometric shapes of the trees [6].

---

[1] †:Corresponding Author

Recently, for the realism in the virtual world, the plausible animation of virtual trees is becoming more important. Therefore, various research efforts are being made for the tree animation. Some of those methods generate the behavior of a virtual tree by animating its branches [7–9] while some others animates the tree by taking only the leaves into account [10, 11].

The method proposed in [11] models the motion of a tree based on the oscillation function of control points. The simulation can be easily performed on GPU in order to represent the forest. However, versatile control over the wind cannot be obtained in this method.

Deformation based animation was also proposed [16]. In this method, the vertices in the tree geometry are translated and rotated in accordance with the external force of the wind and the restoration based on shape matching proposed in [17]. However, this method also suffers from heavy computation when the number of the clusters increases.

3 Tree modeling and deformation

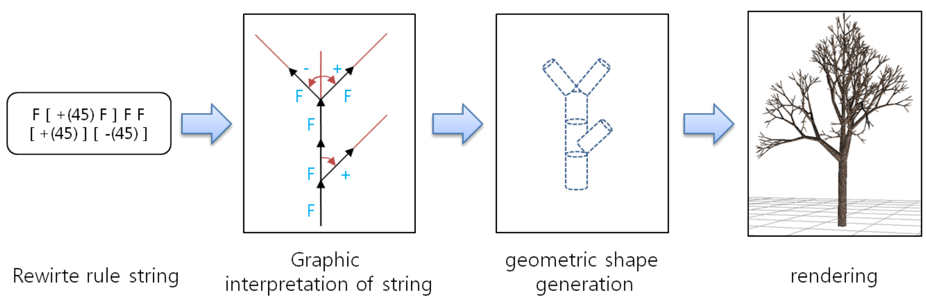

Fig. 1. The process of L-System based tree modeling

We employed the L-system in order to procedurally generate the shape of virtual tree. The L-system employs the rewriting system that modifies the initial string to another string in accordance with mathematical substitution rules. Fig. 1 shows the process of the tree shape modeling based on the L-system.

The deformation techniques have been intensively studied in computer graphics literature. In order to alleviate the computational burden of physical simulation, free form deformation (FFD) approaches have been proposed. However, the deformation based on the FFD cannot produce physically plausible motion. So far, physically plausible and computationally efficient method for realtime animation of complex and realistic virtual tree has not been proposed.

Shape matching method was proposed in[17]. The method focuses on the fact that deformed objects tend to return the original shapes. In the shape matching method, the restoration force can be efficiently computed by finding the optimal linear transformation between the current shape and the original one.

Let us assume, for instance, a deformable object. The set of the vertices of the object is denoted to be $\mathbf{X}$. The original center of mass can be denoted as p_0. If the deformation can be described with a linear transformation $\mathbf{A}$, then the new state $\mathbf{X}'$ can be expressed as follows:

$$\mathbf{X}' = \mathbf{A}\mathbf{X} + p_0 \tag{1}$$

If the object has the exactly the same shape as the original state, the linear transformation should involve only rotation $\mathbf{R}$ and translation $\mathbf{T}$. In this case the new state can be expressed as follows:

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + (\mathbf{T}\mathbf{X} + p_0) \tag{2}$$

where $\mathbf{T}\mathbf{X} + p_0$ is the new center of mass p_1.

Eq. 2 can be considered the target shape that the deformable object should be restored to. In fact, the deformable object undergoes the deformation by the external forces. Therefore, the current state $\mathbf{X}'$ cannot be exactly obtained with $\mathbf{R}$ and $\mathbf{T}$. If we find the $\mathbf{R}$ and $\mathbf{T}$ that optimally transform the original shape close to the current shape, we can obtain the target shape $\mathbf{G}$. Each vertex x'_i should be animated to move back to the goal position g_i in the set $\mathbf{G}$.

Eq. 2 can be rewritten for each vertex x_i as follows:

$$w_i\mathbf{R}(x_i^0 - x_{cm}^0) + x_{cm} - x_i \tag{3}$$

where x_i is the location of the vertex i, and w_i is the weight of the vertex, x_{cm}^0 denotes the center of mass of the object at the initial state while x_{cm} is the current center of mass. The translation transformation $\mathbf{T}$ can be easily computed with Eq. 3. However, $\mathbf{R}$ cannot be easily obtained. Therefore, we first obtain a linear transformation $\mathbf{A}$, and approximate the $\mathbf{R}$ by removing the scaling factors in the transformation.

The optimal linear transformation can be obtained by the least square method as follows[17]:

$$\mathbf{A} = (\textstyle\sum_i m_i p_i q_i^T)(\sum_i m_i q_i q_i^T)^{-1} = \mathbf{A}_{pq}\mathbf{A}_{qq} \tag{4}$$

where $p_i = x_i - t$, and $q_i = x_i^0 - t_0$. Since $\mathbf{A}_{qq}$ is symmetric, $\mathbf{R}$ should be is $\mathbf{A}_{pq}$[17]. The rotation matrix $\mathbf{R}$ can be obtained with polar decomposition of $\mathbf{A}_{pq}$ as follows:

$$\begin{aligned} \mathbf{A}_{pq} &= \mathbf{R}\mathbf{S} \\ \mathbf{R} &= \mathbf{A}_{pq}\mathbf{S}^{-1} \end{aligned} \tag{5}$$

We obtaind $\mathbf{S}^{-1}$ by exploiting the right Cauchy-Green deformation tensor law. The goal location g_i of each vertex can then be described with $\mathbf{R}$ and $\mathbf{T}$.

$$g_i = \mathbf{R}(x_i^0 - t_0) + t \tag{6}$$

The required velocity to easily compute with the goal location with the rigidity control parameter α ranging from 0 to 1 as follows:

$$\begin{aligned} v_i(t+h) &= v_i(t) + \alpha \tfrac{g_i(t) - x_i(t)}{h} + h \tfrac{F^{ext}(t)}{m_i} \\ x_i(t+h) &= x_i(t) + h v_i(t+h) \end{aligned} \tag{7}$$

4 Hierarchical shape matching for realistic animation

In order to avoid the problems of the multi-cluster approach, we propose a hierarchical shape matching. When the goal locations are computed, the hierarchical approach not only performs the local cluster-based shape matching but also employs the global shape matching for preserving the initial shape of the tree.

4.1 Global shape matching

The multi-cluster approach cannot maintain the original shape of the tree since the shape matching forces only considers the shape of each local cluster. In this method, each vertex is forced to move back to the original location of the vertex. The whole shape of the tree can then be easily restored by global shape matching with the adjusted velocity as follows:

$$v_i^{global} = v_i^{local} + k_d(p_i^{original} - p_i) \tag{8}$$

where v_i^{global} is the velocity of i after the global shape matching, and v_i^{local} is the velocity computed in the cluster-based local shape matching. p_i is the location of the particle determined by the local shape matching while $p_i^{original}$ denotes the original location of the vertex. k_d denotes the damping or controlling parameter for the speed of the global restoration.

Fig. 2 shows the concept of the hierarchical shape matching with the local and global shape matchings. Each cluster in a tree model changes its shape when an external force is applied. The local shape matching restores the original shape of each cluster with the waves and oscillations generated. The global shape matching makes the tree more plausible by removing the unrealistic waves and oscillations on the tree.

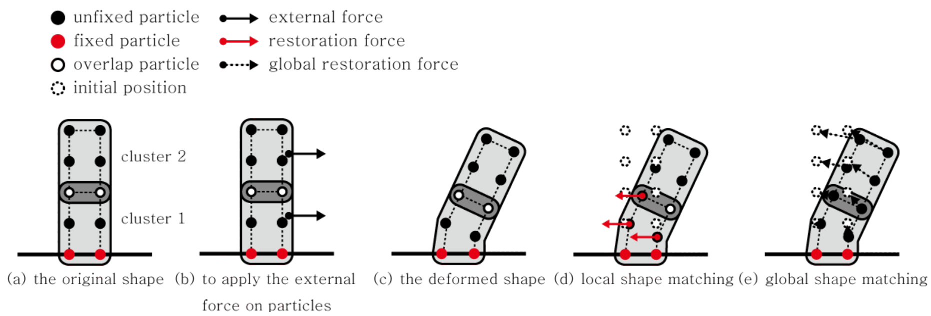

Fig. 2. Hierarchical shape matching concept

4.2 Improved animation based on physical properties

The shape matching method does not take into account the physical properties such as mass. Therefore, in some cases, unrealistic animation is generated where thick branches and thin branches are similarly bended by external forces. However, the thick branches should be less accelerated than thin ones. In order to express such behaviors, we assigned larger masses to the vertices in the thicker branches. If we model every branch with the same number of vertices, the vertices in a cluster with bigger volume can be regarded a sample of the larger area. Therefore, such vertices should be assigned larger masses. Therefore, the mass of each vertex can be computed by considering the volume of the cluster where the vertex is located. V_c, the volume of the cluster of which shape is truncated cone can be easily computed as follows:

$$V_c = \frac{\pi}{3} * (r_t^2 + r_t r_b + r_b^2) * l \tag{9}$$

where r_t and r_b are the radii of the circles at the top and the bottom respectively, and l is the length of the truncated cone. The volume-based mass assignment generates plausible branch animation. Let us assume the volume of the cluster is decreasing in accordance with the height. In that case, the cluster at top-most location will be lighter than others. When an external force is applied to the clusters, the top-most cluster will be accelerated more than any other clusters. After the reasonable animation, the hierarchical shape matching will restore the shape to a plausibly deformed state as shown in the figure.

The real world leaves can be separated from the tree by strong external forces. In our method, the leaves can be easily detached from the tree branches by invalidating the overlapping properties of the overlapping vertices in the leave clusters when the force exceeds a given threshold. The global shape matching will keep the leaves from floating away from the tree. Therefore, we turn off the initial position based matching when the overlapping property of the leave cluster is disabled.

5 Experiment

We implemented the proposed method on a system with Intel 3.47 GHz i7 CPU, 24 G DDR3 RAM, and NVidia GTX 590 GPUs running on Windows 7 OS.

The computational performance in the aspect of frame rates is shown in Table. 1. As shown in the table, even with the complex model with 211,812 vertices, the proposed method could generate an interactive animation with the performance of 32 frames per second.

Table 1. FPS(Frame Per Second)

clusters	particles	FPS
5	300	258
15	900	200
293	13,548	79
867	38,772	62
2,230	99,816	46
4,975	211,812	32

Fig. 3 shows the growing of a virtual tree. The L-system based production rules can generate plausible trees as shown in the figure. By changing the rules or the initial string, we could generate various types of trees. By the nature of the exponential growth, a realistic tree could be easily obtained with a few steps of growth.

Fig. 3. Growth of a virtual tree

The animation result is shown in Fig. 4. For the animation, 211,812 vertices and 4,975 clusters were used. As shown in the figure, the hierarchical shape matching produced a plausible animation where thin branches and leaves move more than thicker branches. The bending tendency of the branches, of course, can be easily controlled in our method.

The leaves could be easily detached from the tree as shown in Fig. 5. As shown in the figure, the detached leaves float in the air and generate realistic wind and tree interaction animation.

Fig. 4. Animation with the hierarchical shape matching

Fig. 5. Detachable tree leaves

6 Conclusion

In this paper, we proposed an efficient and effective animation techniques for virtual trees modeled with L-system. The method proposed in this paper plausibly animates the virtual trees based on hierarchical shape matching (HSM) without any complex dynamic equations. The HSM is an improved version of usual shape matching. In this approach, the object is divided into small clusters, and each cluster is locally animated with the shape matching while the global shape is also adjusted in accordance with the original shape.

The proposed method can be easily exploited to animate various virtual trees, and the computational burden of the proposed method is relatively smaller than those of the traditional approaches such as free form deformation or physically based animation techniques. Moreover, the proposed method generates plausible animation with numerical stability. The method proposed in this paper will be successfully employed in interactive applications such as game.

Acknowledgments. This work was supported in part by the Ministry of Knowledge Economy (MKE), Korea, under the Information Technology Research Center (ITRC) support program supervised by the National IT Industry Promotion Agency (NIPA) (NIPA-2011-(C-1090-1021-0006)), and this work (Grants

No.C0033371) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

References

1. A. Lindenmayer, G. Rozenberg: Developmental Systems and Languages. STOC. 214–221 (1972)
2. M. Aono and T. L. Kunii: Botanical Tree Image Generation. IEEE Computer Graphics and Applications. 55–64 (1982)
3. Weber, Jason and Penn, Joseph: Creation and rendering of realistic trees. SIGGRAPH. 119–128 (1995)
4. Měch, Radomír and Prusinkiewicz, Przemyslaw: Visual models of plants interacting with their environment. SIGGRAPH. 397–410 (1996)
5. Prusinkiewicz, Przemyslaw and Hammel, Mark and Hanan, Jim and Měch, Rdomír: Visual models of plant development. Springer-Verlag New York, Inc. (1997)
6. Yotam Livny and Seoren Pirk and Zhanglin Cheng and Feilong Yan and Oliver Deussen and Daniel Cohen-Or and Baoquan Chen: Texture-Lobes for Tree Modelling. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011). 53:1–53:10 (2011)
7. Jos Stam: Stochastic Dynamics: Simulating the Effects of Turbulence on Flexible Structures. Comput. Graph. Forum. 159–164 (1997)
8. Sakaguchi, Tatsumi and Ohya, Ju: Modeling and animation of botanical trees for interactive virtual environments. ACM (1999)
9. Ding, Lin and Chongcheng, Chen and Liyu, Tang and Qinmin, Wang and Wenqiang, Xu: Interactive Physical Based Animation of Tree Swaying in Wind. IEEE Computer Society (2009)
10. Shin Ota and Machiko Tamura and Tadahiro Fujimoto and Kazunobu Muraoka and Norishige Chiba: Stochastic Dynamics: A hybrid method for real-time animation of trees swaying in wind fields. The Visual Computer. 613–623 (2004)
11. Diener, Julien and Rodriguez, Mathieu and Baboud, Lionel and Reveret, Lionel: Wind Projection Basis for Real-Time Animation of Trees. Computer Graphics Forum. (2009)
12. Manchester, Royaume-Uni: Eurographics Workshop on Computer Animation and Simulation. Springer (2001)
13. Oliapuram, Nimish J. and Kumar, Subodh: Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing. ACM (2010)
14. Y. Akagi and K. Kitajima: Computer animation of swaying trees based on physical simulation. Computers and Graphics. 529–539 (2006)
15. Shin Ota and Tadahiro Fujimoto and Machiko Tamura and Kazunobu Muraoka and Kunihiko Fujita and Norishige Chiba: $1/f^{\beta}$ Noise-Based Real-Time Animation of Trees Swaying in Wind Fields. Computer Graphics International Conference. 1530–1052 (2003)
16. Steinemann, Denis and Otaduy, Miguel A. and Gross, Markus: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association (2008)
17. Müller, Matthias and Heidelberger, Bruno and Teschner, Matthias and Gross, Markus: ACM SIGGRAPH 2005 Papers. ACM (2005)