Capturing, rendering and simulation for large scale grassland

Bin Sheng, *Member, IEEE*

Abstract—Grass is a very import element of nature, however create and implement a large scale, realistic grassland is not that easy due to extremely high computation complexity and large amount of data needed for simulation and rendering. Common sense tells us grass blades are simple, however there are numerous kinds of grass blade in the world and it's not possible for any system to store all kinds of grass blades beforehand. Obtain grass blade with interactive camera can be an intuitive solution for this problem. We provide a method that can obtain grass blade shape with a depth camera, render large scale grass land efficiently and simulation every single grass blade with individual response on the fly.

Index Terms—Grass, Capture, Render, Simulation, GPU

Fig. 1: Test scene

1 Introduction

When an object move through a grass land, every grass blade on the path is pushed aside or run over, and recovers afterward. Individual responses from every single grass blade greatly improve the user experience and increase scene fidelity. With the development of virtual reality(VR) devices and applications, human-computer interaction is paid more attention than ever. Cameras are deployed extensively in recent years, providing more possibility for VR applications. Depth cameras are especially valued since they are able to provide depth information of any scene, which makes it much easier to rebuild any object or scene.

We intend to build a system which is able to obtain grass blade from camera, refine grass blade shape and use it in the large scale grassland scene with high rendering quality as well as realistic simulation response to collision. With a depth camera, we are able to capture grass blade shape instantly instead of using a small number of preset shapes.

In order to achieve real-time performance of rendering and simulation, every grass blade is modeled as a curve with no more than 64 knots. We expand this curve to a triangle strip in the course of rendering. In our system we usually choose 16 as the number of knots for a single curve to guarantee performance. Thus we have to refine the grass shape we capture from camera, and extract the curve which could best describe the grass blade, and calculate the expansion width for each knot in the curve.

Our system is designed to render and simulate a very large scale grassland in which every single grass blade has individual and vivid response to collision. Therefore we apply GPU-based instancing to lower the requirements for memory and bandwidth, and only pay simulation costs when needed. Meanwhile, this is a tile-based rendering and simulation system, no per-tile data is store unless simulation for the tile is required. We implement the method introduced by Han et al. [1] to simulate our grass blade and adopt the moethod introduced by Fan et al. [2] to do tile management.

Our main contribution is to capture grass blade shape with depth camera, shape refinement according to our simulation algorithm, and large scale grass land rendering as well as simulation with high fidelity.

2 Related Work

The most challenging part of grass modeling, rendering and simulation is caused by extremely large quantity. William Reeves and Ricki Blau [3] addressed those challenges in 1985.

Capturing grass blade with camera has not been explored much, however plenty of researches about hair capture have been conducted. There are some similarities between hair and grass blade for their shapes and movement features. Chai et al. [4] uses single image to do hair modeling, they capture hair style from image and are able to render origin hair style at different angle, with different hair materials and change hair style . Afterwards, they extends their work through user's high-level knowledge to get more accurate hair that matches image and is physically real at the same time. By doing it they are able to finish dynamic hair simulation and interactive hair style editing, which make it possible to apply this hair manipulation in video [5]. A structure-aware hair capture method [6] was introduced in 2013, they adopt a method to generate hair strand segments, set up a connection graph to guarantee hair growth to areas with missed geometry information and connect hair strands with consistent curvature. A method to capture hair using simulated examples was introduced by Hu et al. [7], they use simulated samples as reference to generate hair, and this method could be applied with unconstrained and constrained hair. They also came up with a method to capture braided hair style [8]. They adopt a data-driven reconstruction and use procedurally generated examples to fit captured hair strands. Xu et al. introduced a space-time optimization method to capture dynamic hair. This method could faithfully capture hair strands' shape as well as spatial details.

In previous works, grass is moved aside when interaction between grass and object happens [9]. Spring-mass system is also used to model grass blade and simulate grass-object interaction [10] [11]. We adopt the simulation algorithm introduced by Han et al. [1]. This method treats collision as hard constraint, meanwhile treat length, bending and twisting as soft constraints in the iterative solver for grass-object interaction. We employ the rendering and simulation framework introduced by [2]. With this frame work, we are able to perform collision computation on GPU and do grass blade instancing on the fly. We implement our capture module on the basis of this framework to obtain more accurate and diverse grass types.

3 Algorithm Overview

4 Blade Capture

5

6 Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgments

The authors would like to thank...

References

[1] D. Han and T. Harada, "Real-time hair simulation with efficient hair style preservation," pp. 45–51, 2012.

[2] Z. Fan, H. Li, K. Hillesland, and B. Sheng, "Simulation and rendering for millions of grass blades," in *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*. ACM, 2015, pp. 55–60.

[3] W. T. Reeves and R. Blau, "Approximate and probabilistic algorithms for shading and rendering structured particle systems," in *ACM Siggraph Computer Graphics*, vol. 19, no. 3. ACM, 1985, pp. 313–322.

[4] M. Chai, L. Wang, Y. Weng, Y. Yu, B. Guo, and K. Zhou, "Single-view hair modeling for portrait manipulation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 116, 2012.

[5] M. Chai, L. Wang, Y. Weng, X. Jin, and K. Zhou, "Dynamic hair manipulation in images and videos," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 75, 2013.

[6] L. Luo, H. Li, and S. Rusinkiewicz, "Structure-aware hair capture," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 76, 2013.

[7] L. Hu, C. Ma, L. Luo, and H. Li, "Robust hair capture using simulated examples," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 126, 2014.

[8] L. Hu, C. Ma, L. Luo, L.-Y. Wei, and H. Li, "Capturing braided hairstyles," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 225, 2014.

[9] S. Guerraz, F. Perbet, D. Raulo, F. Faure, and M.-P. Cani, "A procedural approach to animate interactive natural sceneries," in *Computer Animation and Social Agents, 2003. 16th International Conference on*. IEEE, 2003, pp. 73–78.

[10] K. Chen and H. Johan, "Real-time continuum grass," in *Virtual Reality Conference (VR), 2010 IEEE*. IEEE, 2010, pp. 227–234.

[11] N. Hempe, J. Rossmann, and B. Sondermann, "Generation and rendering of interactive ground vegetation for real-time testing and validation of computer vision algorithms," 2013.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.