REAL-TIME HAIR RENDERING WITH SCREEN SPACE ADPATIVE LEVEL OF DETAIL

Markus Rapp, Simon Spielmann*

*Institute of Animation – Filmakademie BW, Germany, research@filmakademie.de

Keywords: hair, real-time, level of detail, tessellation

Abstract

We present an approach to render hair by using a small number of guide strands to generate interpolated hairs on the GPU. Each hair strand is composed by segments, which can be further subdivided to render smooth hair curves. The appearance of the guide hairs as well as the size of the hair segments in screen space are used to calculate the amount of detail, which is needed to display smooth hair strands.

1 Introduction

With the increased performance of newer graphic cards, the rendering of hair in form of single hair strands became possible. The tessellation pipeline of the GPU allows to generate up to 64 hairs out of one hair guide and further subdivide each hair segment up to 64 times. The number of generated hairs is described with a tessellation factor, while a detail factor defines how many times a hair segment has been subdivided. Therefore, the detail factor describes the smoothness of the hair.

2 Previous Work

Tariq [2] implemented real-time hair rendering using the tessellation pipeline. In her work, 166 guide hairs were used to generate interpolated hair with a single strand or a multi strand interpolation pattern. Additional work about real-time hair rendering was done by [1] with the implementation of TressFX, which was used for the Crystal Dynamics reboot of the video game Tomb Raider. TressFX main focus is performance. However, both implementations do not concentrate on the smoothness of the hair strands. In Tariq's hair demo the number of subdivisions of a hair segment is the same for all guide hair segments and cannot be changed at run-time. On its part, TressFX does not support hair segment subdivisions and rendering artefacts are visible if the hair takes up the whole screen.

3 Implementation

Our idea is to have a dynamic detail factor for each hair strand segment, which is dependent on the shape of the hair and the size of a hair segment in screen space. The hair rendering is based on Tariq's multi strand rendering. The difference, however, is the calculation of the detail factor in the hull/control shader, which is called once per hair segment. First, back-face culling is performed. Afterwards, the two vertices of the hair segment are transformed into screen-space to get the size of the hair segment. According to this size the detail tessellation factor is calculated. Additionally, the form of the hair strand is taken into account for the calculation of the detail factor. As pre-calculation step, the angle between the previous segment and the current segment as, well as the angle between current segment and the next segment is calculated. The larger angle is then used in the shader to influence the final detail factor.

4 Results

The main difference of the rendered hair in Figure 1 is the frame rate in which the hair is rendered on a NVidia Geforce GTX 580. The frame rate is above 60 fps, without subdivision of the hair segments. The hair is rendered with only 1 fps when the detail factor is set to 64. The screen space adaptive level of detail (SSALOD) version renders with 42 fps.

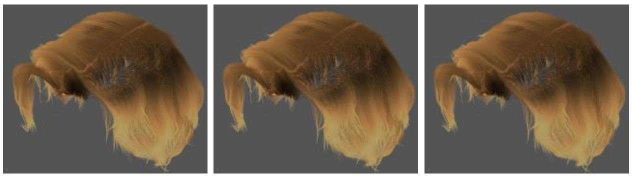

Figure 1: Hair with detail factor 1(left), detail factor 64 (middle) and SSALOD (right)

Figure 2 shows how subdivision of hair segments is performed. The straight part of the hair are less subdivided then the curvy parts.

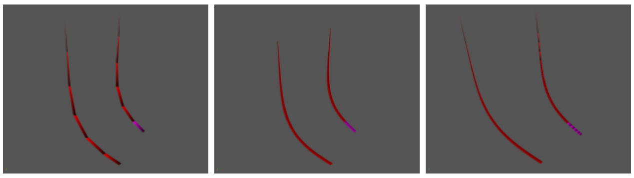

Figure 2: Debug view of hair segments with detail factor 1(left), detail factor 64 (middle) and SSALOD (right)

5 Conclusion

Our approach allows smooth hair shapes at close up views for curvy hair and at the same time reduced level of detail straight hair at a farer distance.

Acknowledgements

This work has been carried out in the German Research Foundation (DFG) funded project SARA (AR 892/1-1).

References

[1] T. Martin, E. Woflgang, N. Thibieroz, J. Yang, J. Lacroix. "TressFX: Advanced Real-Time Hair Rendering", *GPU Pro*, **5**, pp. 193-209, (2014).

[2] S. Tariq. "Real-Time Hair Rendering on GPU", *SIGGRAPH 2008*, (2008).