Assignment 2 : Simulation of Hair Motion with Moving Head

NAME: JUST

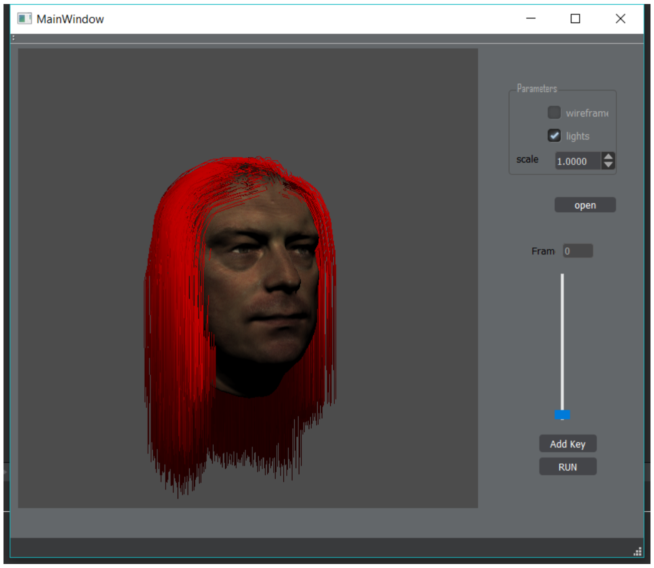
1 INTRODUCTION

In this assignment, I have implemented a simple physical simulation program to demonstrate the dynamics of human hair. The system allows users to interact with the head model, and shows the simulation of the hair. The features of the system are listed below:

(1) A graphics program consists of animation controlling widgets to create key-frame animation.
(2) Interactive simulation.
(3) Hair generation and simulation.
(4) Head-hair collision detection and solution.
(5) Basic rendering of hairs.

2 IMPLEMENTATION DETAILS

As assignment 1, I create the system(as shown in Figure 1) by Qt 5.8.0.



☐ wireframe
☑ lights
scale 1.0000
Fram 0

Fig. 1. The interface

2.1 Hair generation and simulation

To simulate the animation, I produced the class ***Hair*** to manage the whole process including initialization and update, ***Hairline*** to handle the guide strands, and ***Spring*** to implement the mass-spring system.

Generation. To find out where to generate the hair strands, I paint the hair area in black(as shown in Figure 2). When loading the head model, the system is able to save the hair candidates vertices. Then the system will sample dozens of vertices as the root positions of the guide strands.

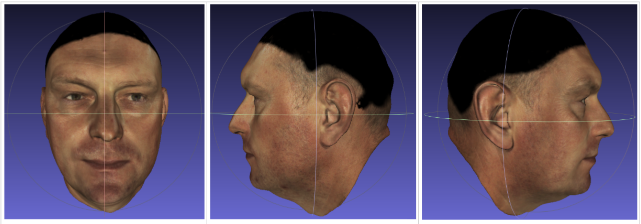


Fig. 2. The painted head model

Initialization. In the initial stage, the system generates ***GUIDEHAIRNUM_JUST*** guide hair strands and ***SPRING_NUM_JUST*** springs in every strand. They are all planted along with the normal of the root vertex in a gap of ***SPRING_LENGTH_JUST***.

Simulation. Every mass point of the guide strand is forced by the forces of its neighbor mass points $F_{spring} = k(x - x_0)$, gravity, damping force and the force from the head(if there is a head-hair collision). The system calculates the motion of every root point(keep rigid with the head) and then transfer the motion to the neighbor point. To avoid the spring too long, there is a length constrain in each update(***Spring::checkSpringLength***).

2.2 Head-hair collision detection and solution

Instead of using a capsule bounding box(Figure 3 left), I build a BvH tree[2] for the head model to detect the head-hair collision(Figure 3 right , code in ***Spring::detectCollision***). After the simulation, the system generates a ray, starting from head center c to the updated point x, to shoot the head mesh and calculates the intersection point x' in the mesh. When the collision is detected($distance(x' - c) > distance(x - c)$), the strand will get a additional force from the head $F_{head} = k_{head}(x - x')$. And it will push the strand out of the head.

2.3 Basic rendering of hairs

After the previous steps, the system has got a physical simulation, but the hair strands are broken lines and the number of guide strands is not enough to simulate a realistic hair dynamics. The system will smooth the hair strands and create more hair strands by Archimedean spiral algorithm.

$$x = (a + b\theta)cos(\theta), y = (a + b\theta)sin(\theta) \tag{1}$$

Finally, the system renders the hair strands by Kajiya-Kay shading model[1](Figure 4 ***hair.frag***).

Computer Animation & Physical Simulation Course Assignment

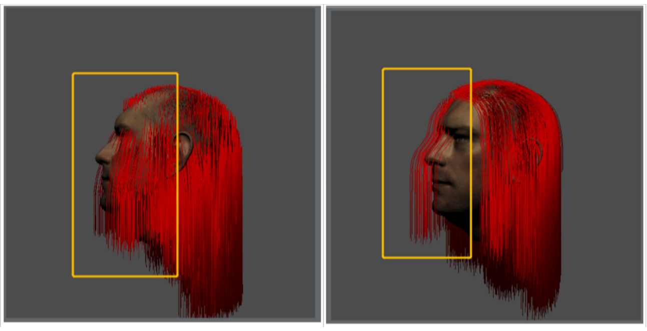


Fig. 3. Collision results – capsule model collision detection result(left figure), Bvh tree collision detection result(right figure)

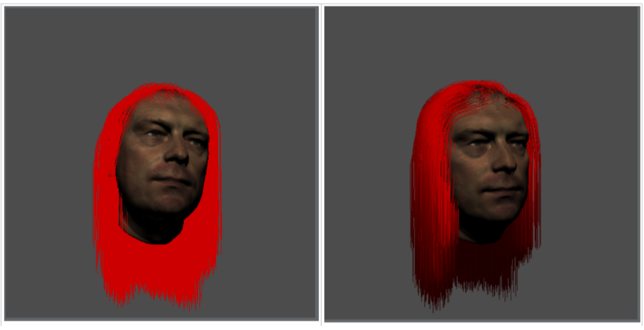


Fig. 4. No Kajiya-Kay shading(left), Kajiya-Kay shading(right)

3 RESULTS

Please watch the demo video to see more features and results.

REFERENCES

[1] James T Kajiya and Timothy L Kay. Rendering fur with three dimensional textures. *international conference on computer graphics and interactive techniques*, 23(3):271–280, 1989.

[2] Marek Vinkler, Vlastimil Havran, and JiÅŹi Sochor. Technical section: Visibility driven bvh build up algorithm for ray tracing. *Computers Graphics*, 36(4):283–296, 2012.

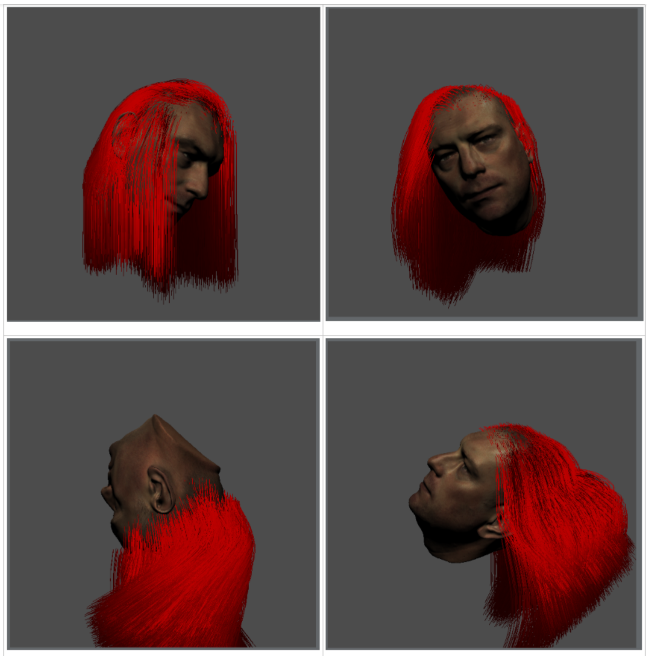


Fig. 5. Results