织物渲染

罗幸荣

复现工作

结果展示

展望

复现工作

复现工作：目标论文

Real-time Fiber-level Cloth Rendering

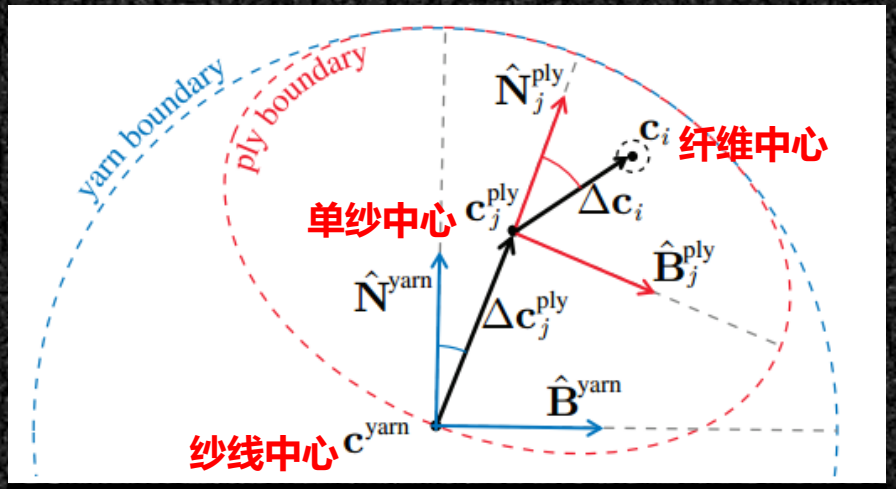
Kui Wu*
University of Utah

Cem Yuksel†
University of Utah

Figure 1: ***Examples of rendering fiber-level cloth at real-time frame rates:*** *A sweater model that consists of 356K yarn curve control points and over 20M fiber curves, rendered using different yarn types with different fiber-level geometry. Notice the difference in appearance.*

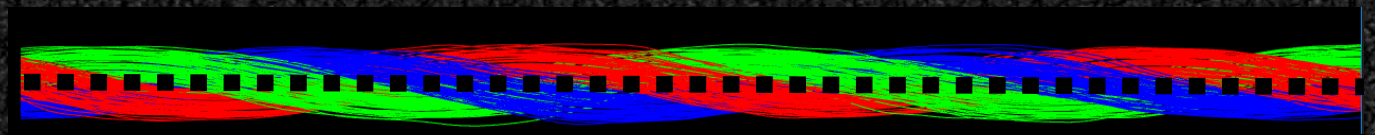
在GPU中对纤维级织物进行生成并实时渲染

在CPU中对纱线几何模型进行生成



几何关系

纤维围绕单纱中心，单纱围绕纱线中心。给定z轴上的控制点，生成常规纤维如下：

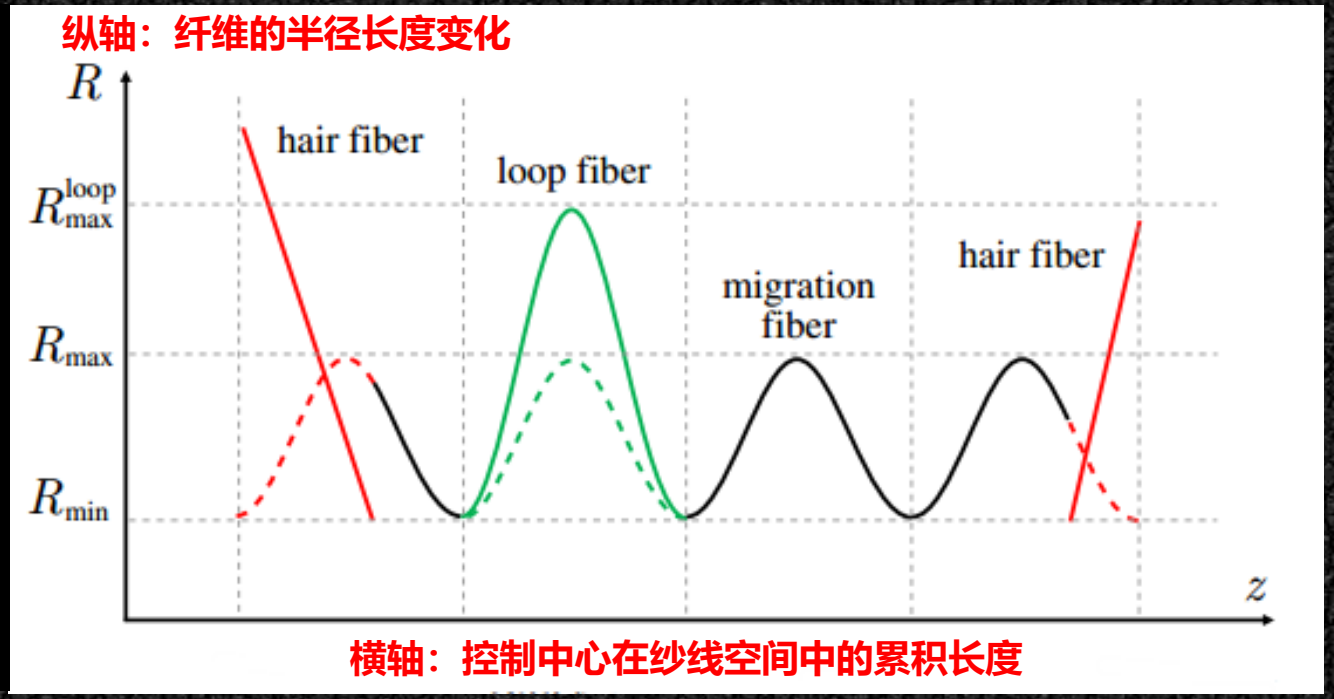


在纱线空间中(z轴)生成纱线

在CPU中对纱线几何模型进行生成

自由纤维与环形纤维可以在常规纤维生成过程中对任意完整的周期进行选取，并改变单纱半径长度来进行仿真。其实现的关键在于周期的选择，如下所示：

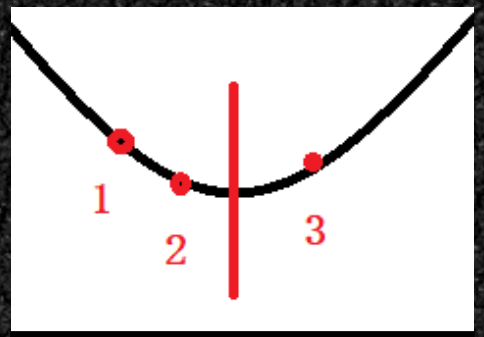
纤维半径变化曲线



在CPU中对纱线几何模型进行生成

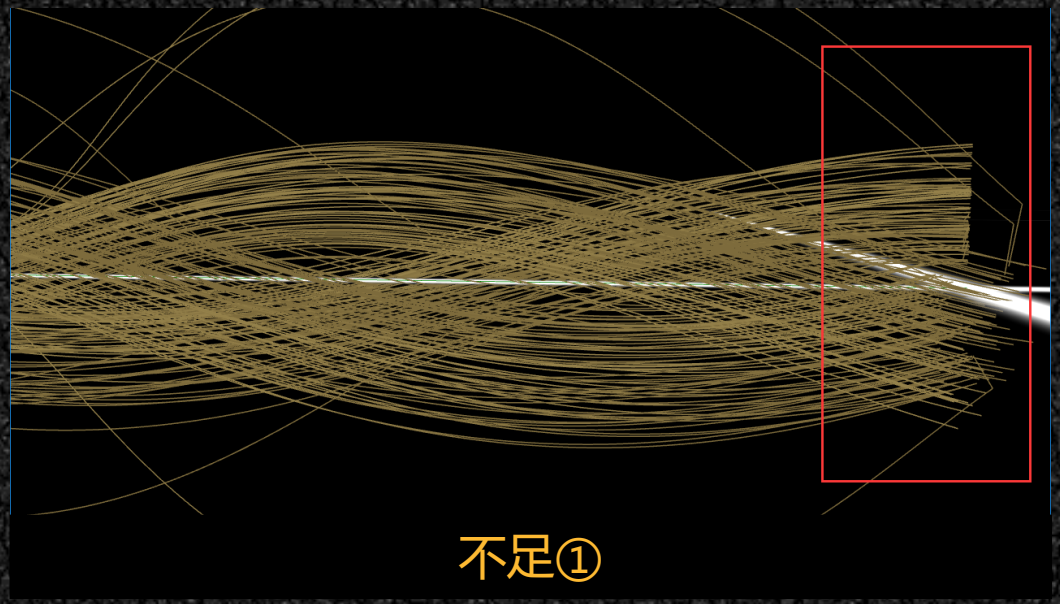
$$R_j(\theta) = \frac{R_{ji}}{2}(R_{\max} + R_{\min} + (R_{\max} - R_{\min})\cos(\theta_{ji} + s\theta))$$

常规纤维的半径按余弦函数进行变化，师兄通过余弦值三点作差来对波谷进行判断。如下图所示，若(cos1-cos2)*(cos2 - cos3) < 0则说明该点为极值点，但是可能是波峰也可能是波谷，再用cos3 < 0加以约束即可对波谷进行锁定，并认为从此开始一个新的周期，从而通过随机函数选择该周期的纤维种类，并设置相应的半径：

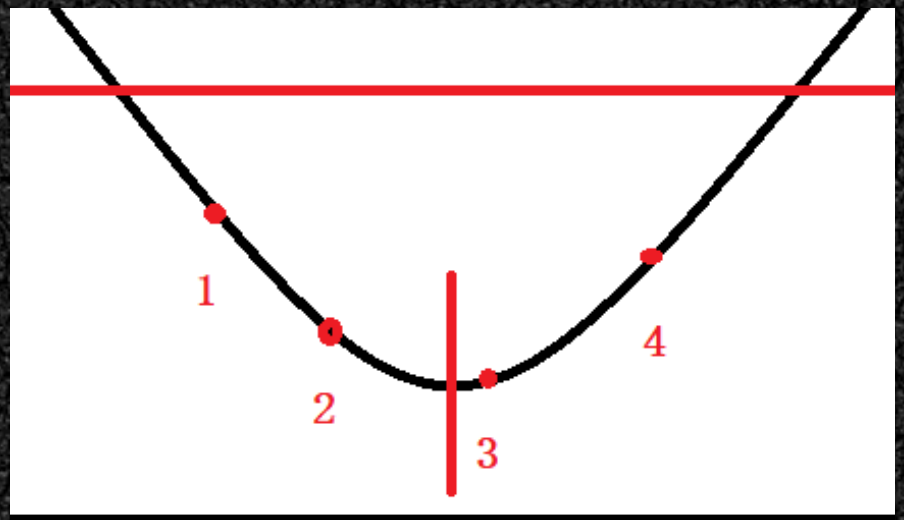


余弦函数波谷

在CPU中对纱线几何模型进行生成



不足①

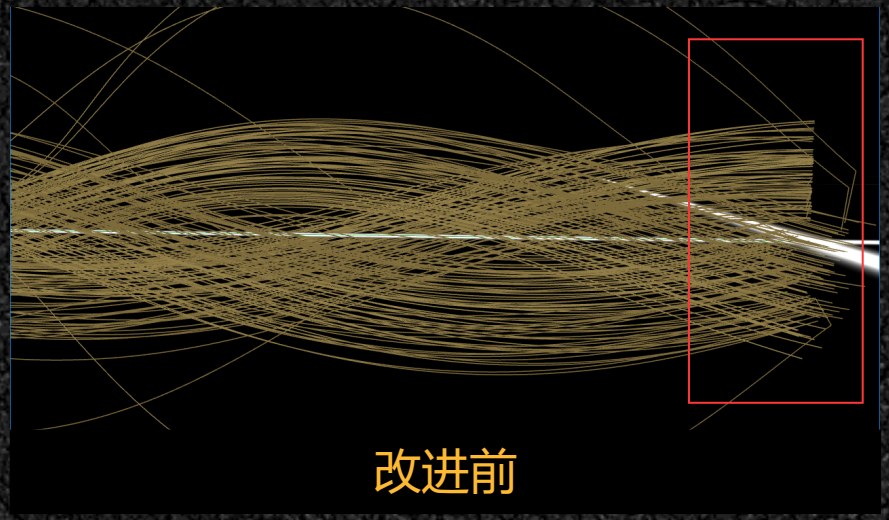


不足②

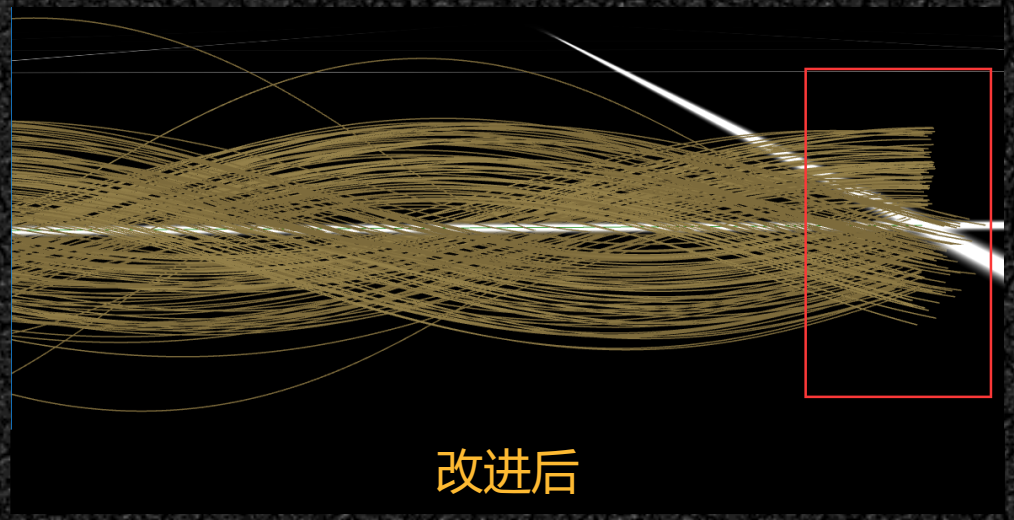
① 由于需要三点作差，导致初始纤维点难以处理，造成非常规纤维曲线突变弯曲。

② 当控制点的离散间隔较大时，三点作差无法准确的对周期进行判断，如上图所示，其中的3将不会被判定为属于新的周期。

在CPU中对纱线几何模型进行生成



改进前

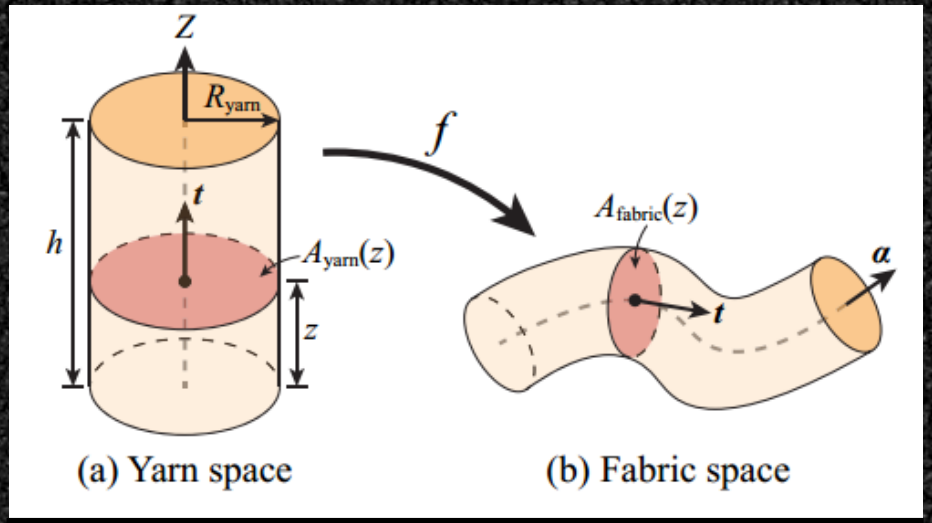


改进后

周期的判断并没有那么复杂。所有纤维的初始角度都会被定义到[-PI, PI]，故直接根据纤维的角度θ即可进行求解。初始周期必然为0，而后根据前后端点所属周期是否一致：

$$cyc = (\theta + PI) / (2 * PI)$$

在CPU中对纱线几何模型进行生成



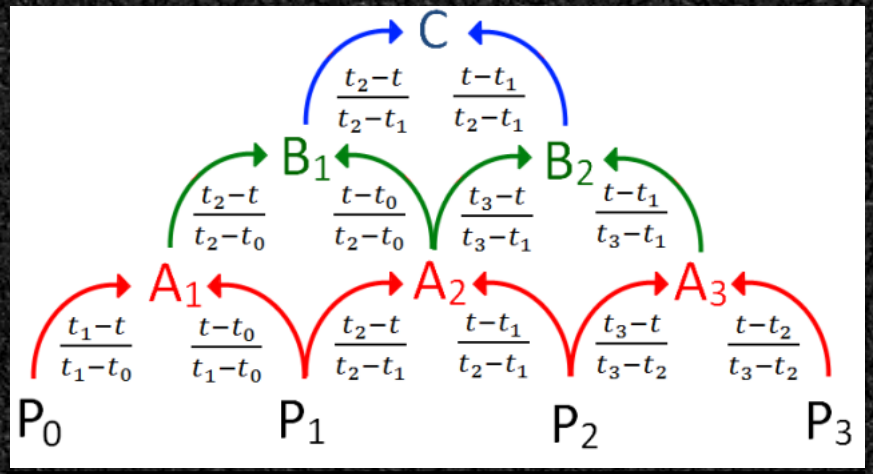
纱线空间与世界空间示意图

纱线空间中的类圆柱体纤维集合到世界空间中的指定纱线控制点，关键在于对样条曲线插值点的世界坐标进行计算，以及对所在位置的正交坐标系(reference frame)进行建立，进而通过下式对纱线空间中的纤维坐标进行映射。

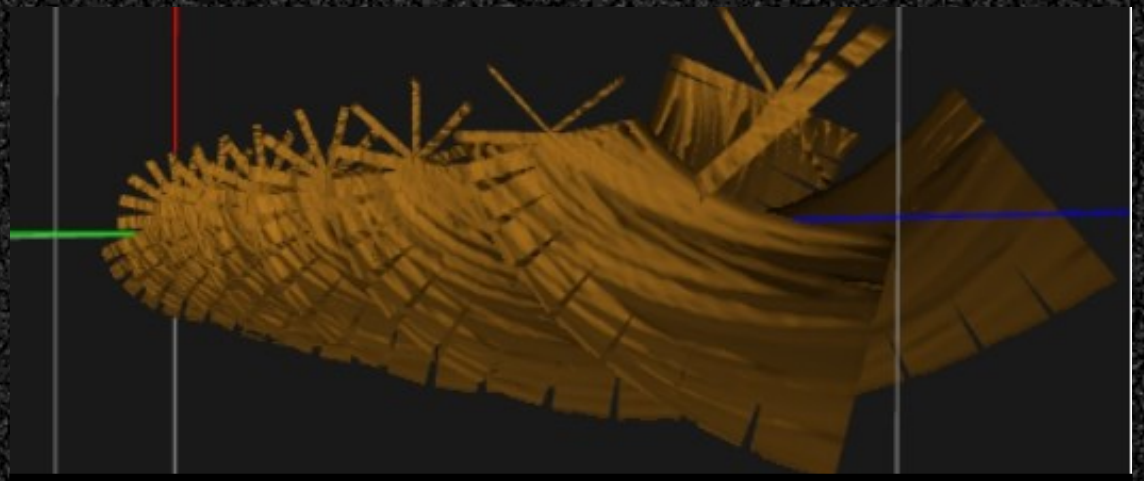
$$f([x,y,z]) = x\,\boldsymbol{n}(z) + y\,\boldsymbol{b}(z) + \boldsymbol{\alpha}(z)$$

在CPU中对纱线几何模型进行生成

因纱线模型BCC文件的格式要求，师兄在论文中采用了向心卡特摩尔罗样条曲线进行纱线控制点进行插值，样条曲线通过四个控制点递归进行插值，没有显式的参数方程，无法进一步对各点斜率、曲率等进行求解，这使得师兄需要将两个插值点之间的连线作为近似正切向量，是导致面片衔接出现问题的直接原因：



向心卡特摩尔罗样条曲线递归过程



近似正切向量带来的面片断层问题

在CPU中对纱线几何模型进行生成

康奈尔大学在2017《Fiber-Level On-The-Fly Procedural Textiles》 提到，为了圆柱曲线在空间变换的过程中局部Frame的旋转更加平滑，他们采用cubic Hermite splines来对曲线进行表示。通过对wiki的学习了解到，cubic Hermite splines只需要起止控制点以及相应的正切向量即可插值计算，参数方程如下所示：

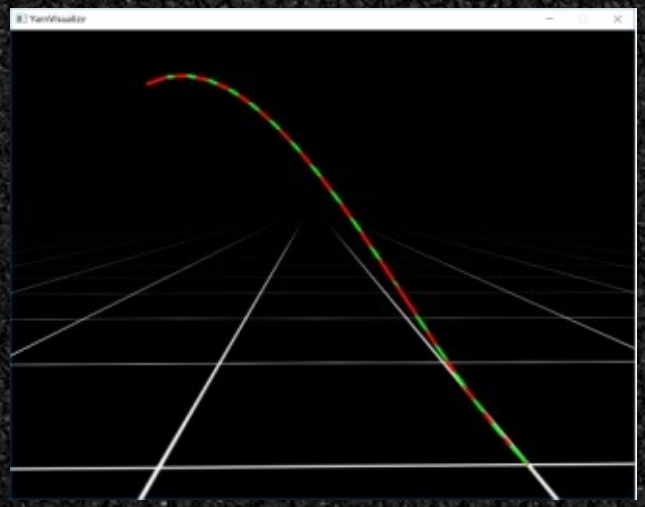
$$\boldsymbol{p}(t) = (2t^3 - 3t^2 + 1)\boldsymbol{p}_0 + (t^3 - 2t^2 + t)\boldsymbol{m}_0 + (-2t^3 + 3t^2)\boldsymbol{p}_1 + (t^3 - t^2)\boldsymbol{m}_1$$

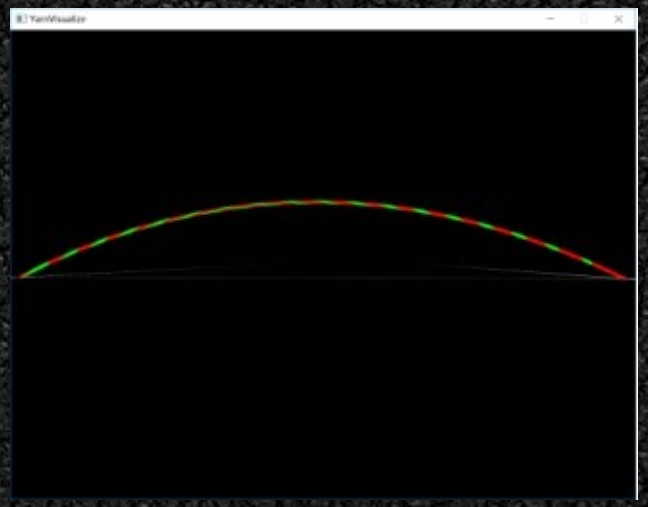
对于给定的控制点，实现CHS样条曲线的关键在于正切向量的选择策略。起始控制点的正切向量满足如下关系时，cubic Hermite splines即为Catmull-Rom splines：

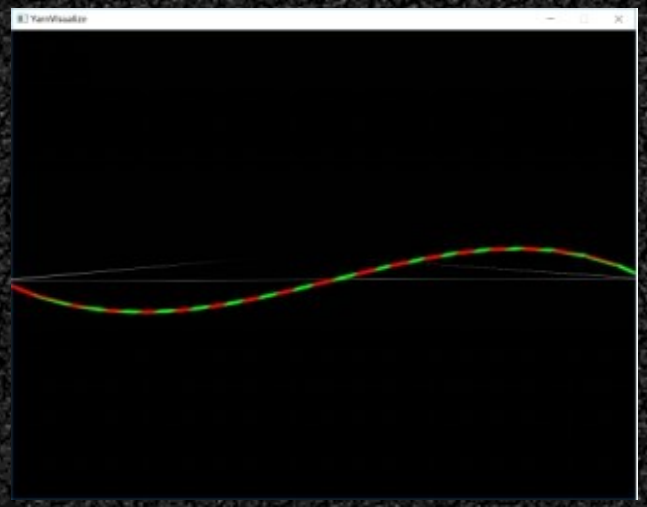
$$\boldsymbol{m}_k = \frac{\boldsymbol{p}_{k+1} - \boldsymbol{p}_{k-1}}{t_{k+1} - t_{k-1}}$$

在CPU中对纱线几何模型进行生成

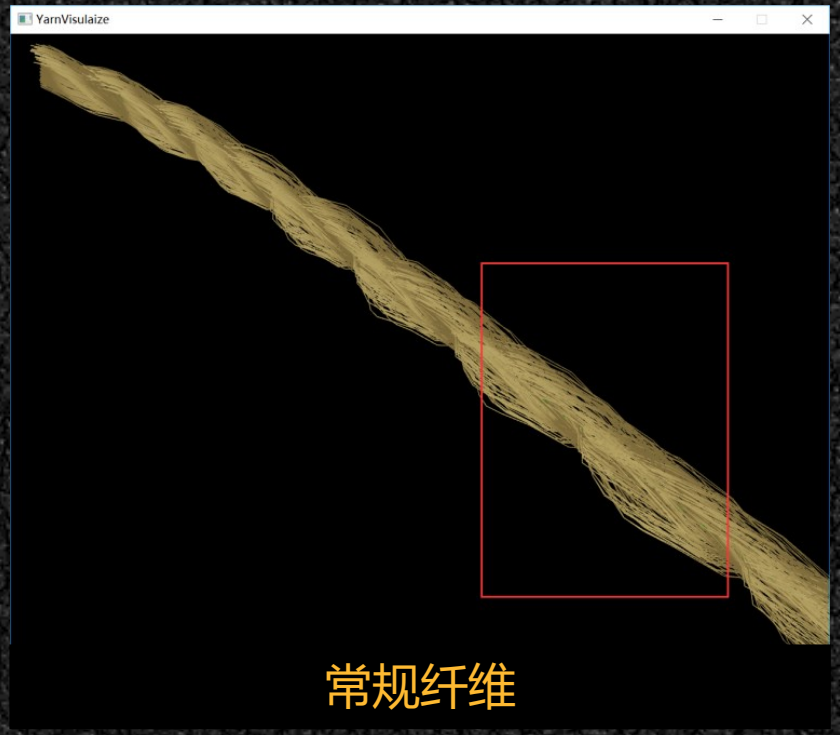
Catmull-Rom曲线是特殊的cubic Hermite splines曲线，而犹他大学的Bcc纱线模型需要使用Catmull-Rom样条曲线进行插值，故选择CHS样条曲线不仅可以得到CRS样条曲线，且可以得到完整的三次曲线参数方程。在程序中同时对CHS和CRS样条曲线进行了实现。经过测试可以发现，上述取值的CHS与uniform的CRS确实是相一致的(红色为CHS，绿色为CRS)：



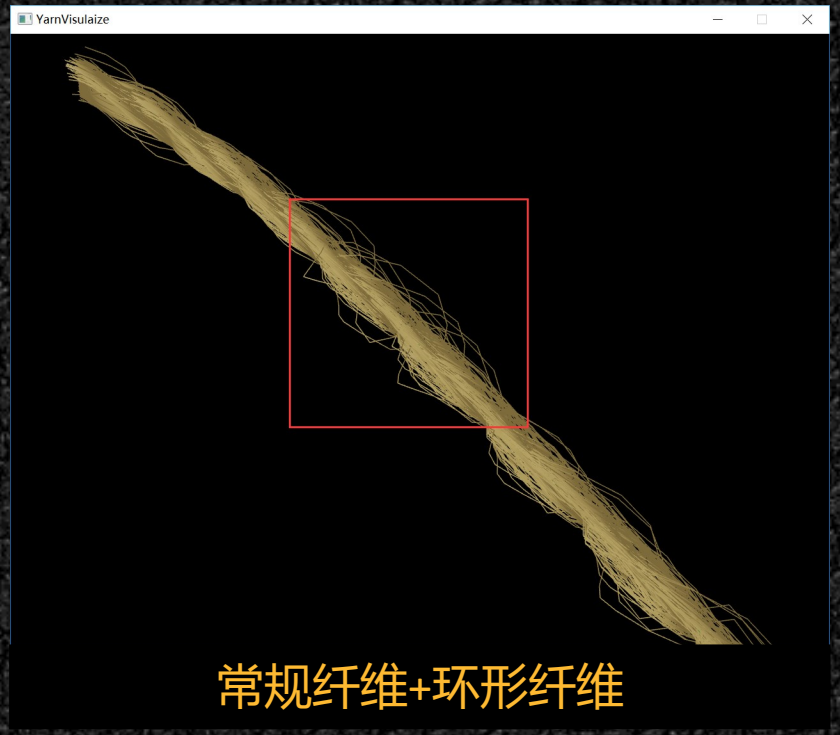




在CPU中对纱线几何模型进行生成



常规纤维



常规纤维+环形纤维

罗德里格旋转公式是计算三维空间中一个向量绕旋转轴旋转给定角度得到新向量的计算公式。师兄直接依据纱线空间的正切向量与世界空间中的近似正切向量得到了变换矩阵。当控制点曲线较为复杂时，曲线杂乱。

在CPU中对纱线几何模型进行生成

$$P = \mathbf{A}t^3+\mathbf{B}t^2+\mathbf{C}t+\mathbf{D}$$

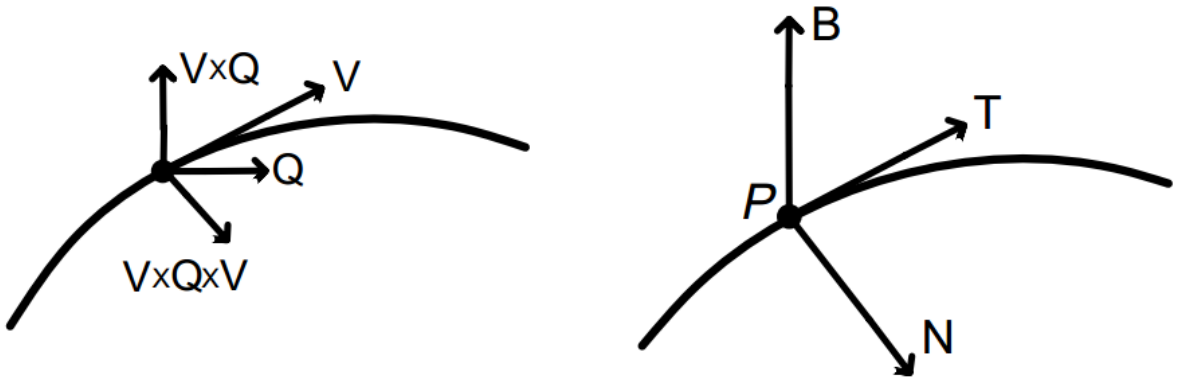
正确的方法是基于曲线参数方程建立reference frame。以上述曲线参数方程为例，对Frenet frame进行建立如下：

$\mathbf{V} = 3\mathbf{A}t^2+2\mathbf{B}t+\mathbf{C}$.

The principal normal is often defined to be in the direction of curvature, $\mathbf{K} = \mathbf{V}\times\mathbf{Q}\times\mathbf{V}/|\mathbf{V}|^4$.

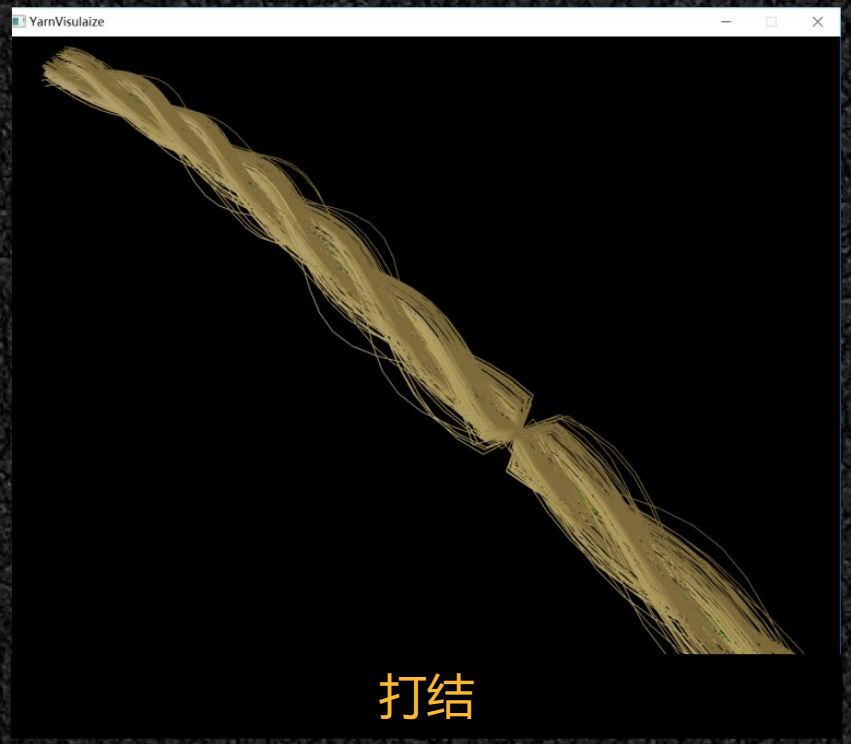
Q is the acceleration of the curve, *i.e.*, the derivative of velocity, $6\mathbf{A}t+2\mathbf{B}$. Thus,

$\mathbf{T} = \mathbf{V}/|\mathbf{V}|$, $\mathbf{N} = \mathbf{V}\times\mathbf{Q}\times\mathbf{V}/|\mathbf{V}\times\mathbf{Q}\times\mathbf{V}|$, and $\mathbf{B} = \mathbf{T}\times\mathbf{N}$.



在CPU中对纱线几何模型进行生成

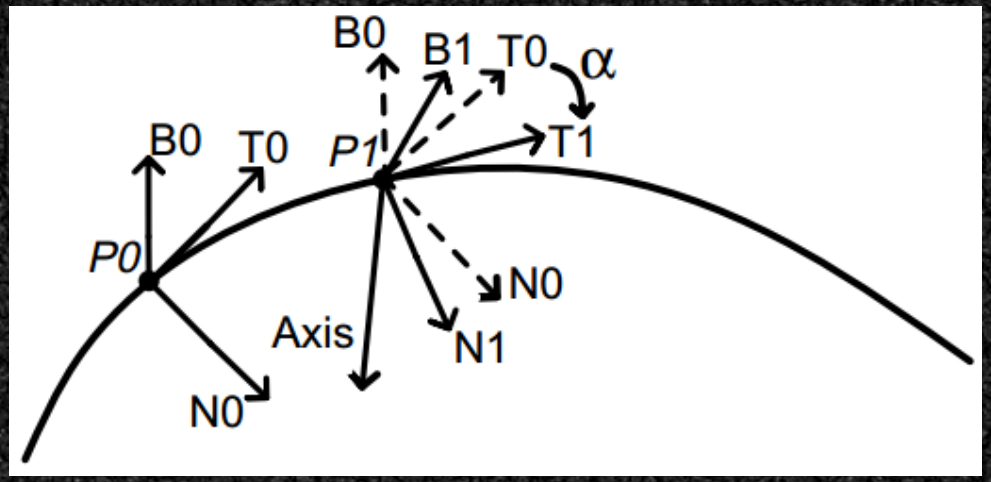
经实现后发现，Frenet Frame在曲率存在的情况下工作良好，当曲率不存在时，会生成打结或断层的纤维，但值得注意的是曲线杂乱、不平滑的问题已经得到很大的改善。



打结

断层

在CPU中对纱线几何模型进行生成



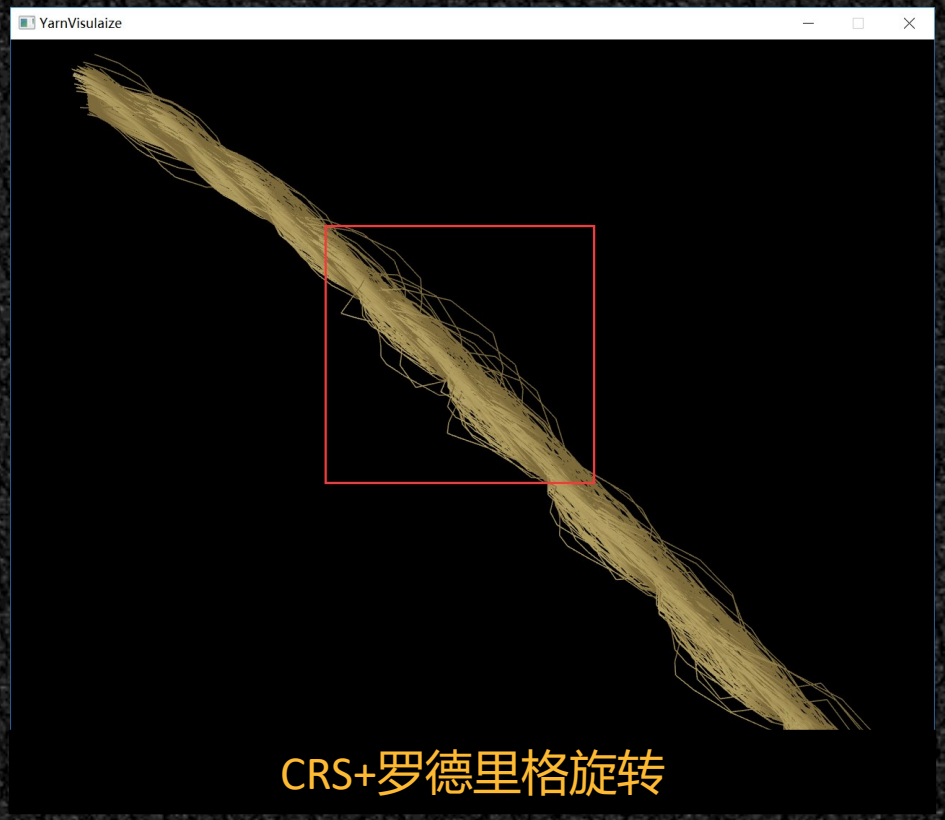
曲率消失的问题可以通过局部frame旋转来进行解决，即旋转起始点的frame来得到后续插值点的frame：

① 对于曲率存在的起始纱线控制点，按Frenet Frame进行计算。

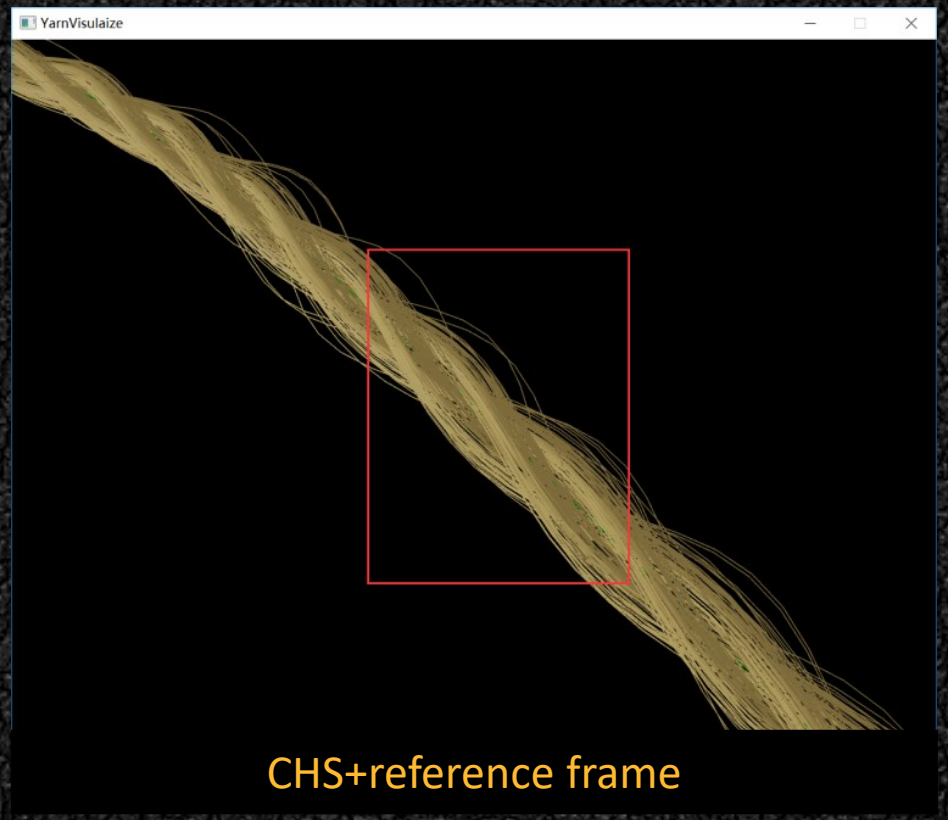
② 对于曲率不存在的起始纱线控制点，将N指定为任意与T垂直的向量均可。

因为需要以前后点正切向量的叉乘为旋转轴进行旋转，若前后正切向量相同，则不存在旋转轴，但只需直接赋值传递即可。实现时需要注意浮点精度。

在CPU中对纱线几何模型进行生成

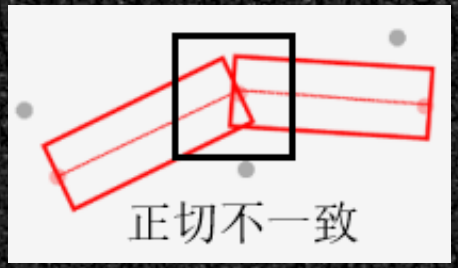


CRS+罗德里格旋转



CHS+reference frame

在GPU中对纱线几何模型进行生成：核心纤维

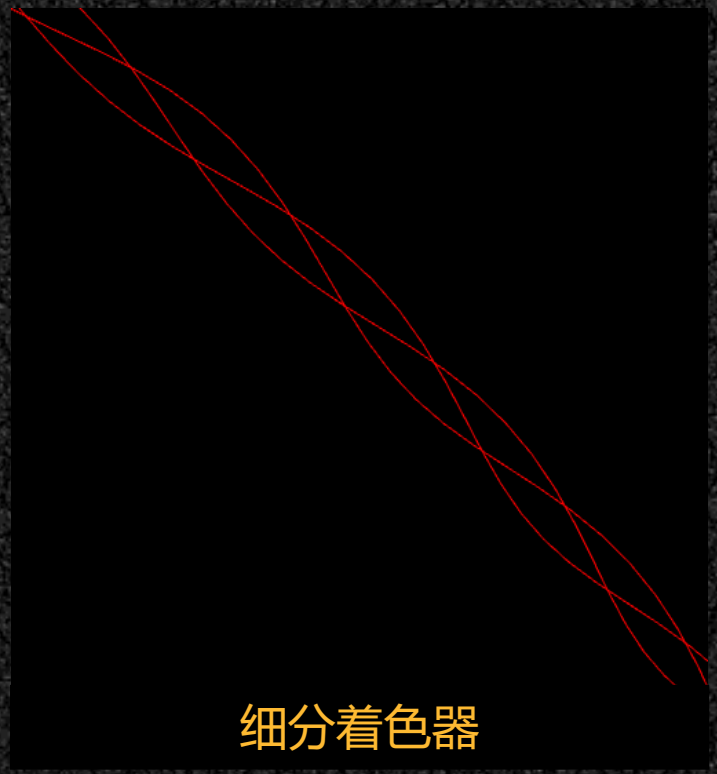


向心卡特摩尔罗曲线与罗德里格旋转公式造成正切向量不一致，使得纱线中心不一致，单纱中心不一致，进而面片无法衔接。师兄通过适当补偿长度和宽度来解决上述问题，但随着视点近距离的移动，将依然呈现出错误的结果。

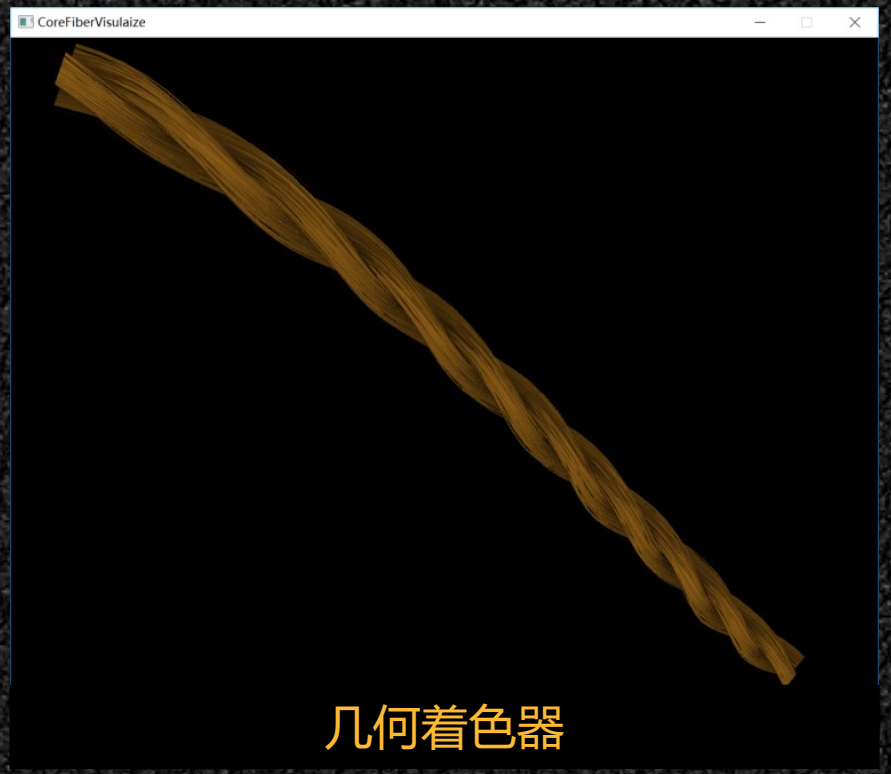
Cubic Hermite Spline显式的参数方程使得可以对任意插值点的正切向量进行计算，而不需要通过相邻插值点作差来进行计算，这就不会导致插值点在前后核心贴图计算中的不一致，从而彻底解决问题。

在GPU中对纱线几何模型进行生成：核心纤维

- 对于输入的纱线中心控制点，细分着色器将纱线一分为三，加入几何着色器通过公告牌技术将对等值线扩展为始终朝向视点的核心纤维面片。



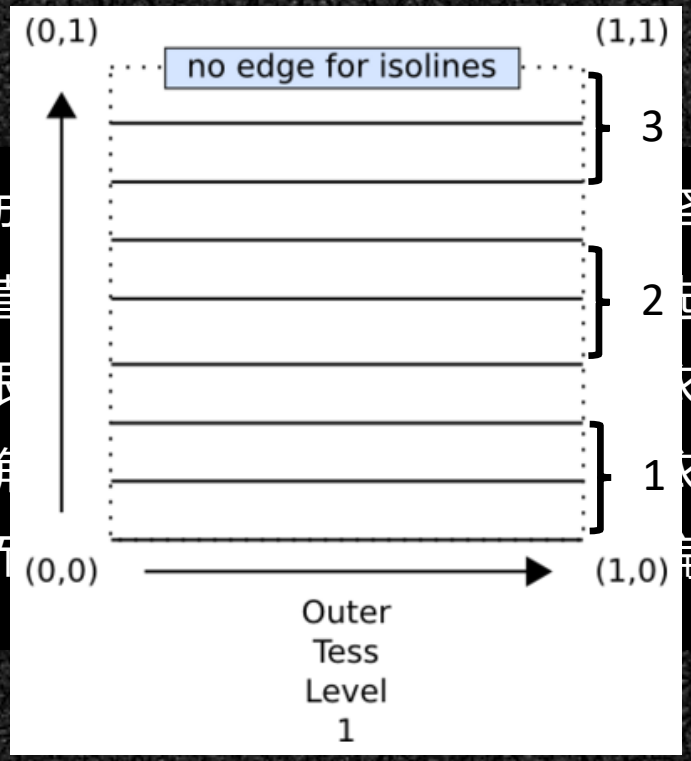
细分着色器

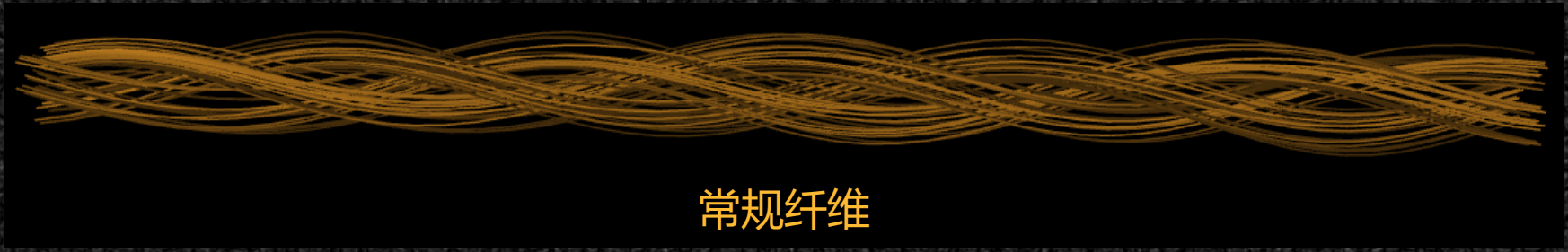


几何着色器

在GPU中对纱线几何模型进行生成：非核心纤维

非核心纤维在常规纤维生成[illegible]择，首先需要在GPU中对常规纤维进行生成。细分着色器的等值[illegible]选择的纱线由三根纱线组成，故将一根纱线中心连线细分为63根[illegible]依次分配给不同的单纱：首先需要对当前纤维所属的单纱进行确[illegible]依据当前纤维的初始旋转角度与半径，在单纱中心的基础上进而[illegible]偏移进行计算：

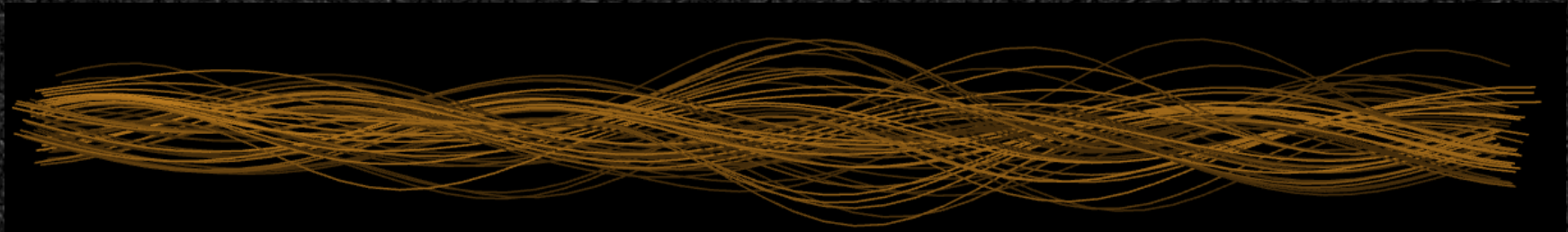




常规纤维

在GPU中对纱线几何模型进行生成：非核心纤维

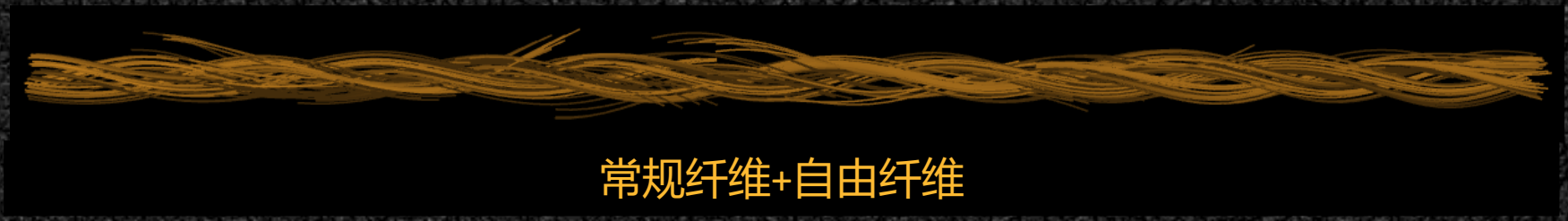
在程序中对纤维周期所属的类型通过随机数进行确定。不同类型纤维的区别在于最大半径R_max，故直接对周期相应的半径进行存储，并作为一维纹理传入着色器，使得细分计算着色器可以避免分支判断直接对环形纤维进行生成：



常规纤维+环形纤维

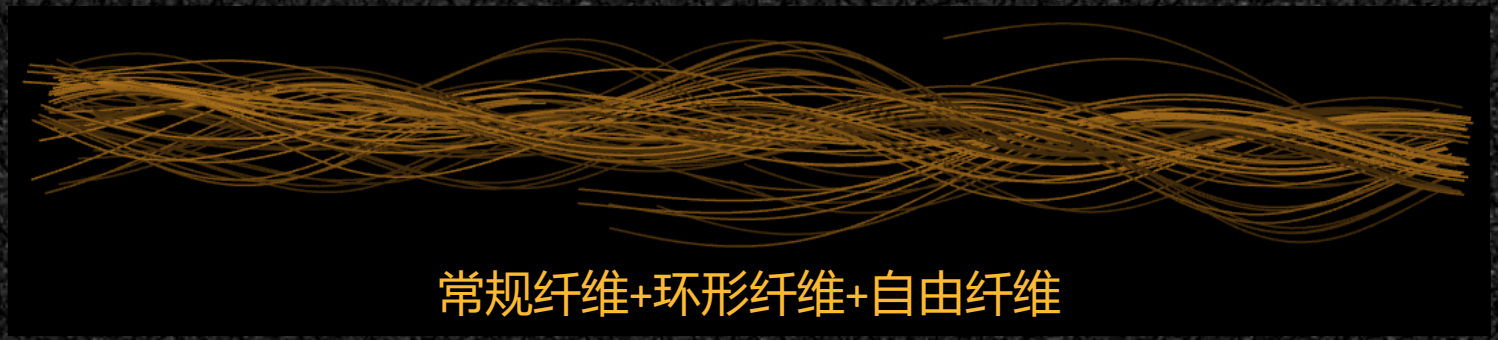
在GPU中对纱线几何模型进行生成：非核心纤维

- 自由纤维由于其半径在单个周期中呈线性变化，所以对自由纤维的分支判断不可避免：



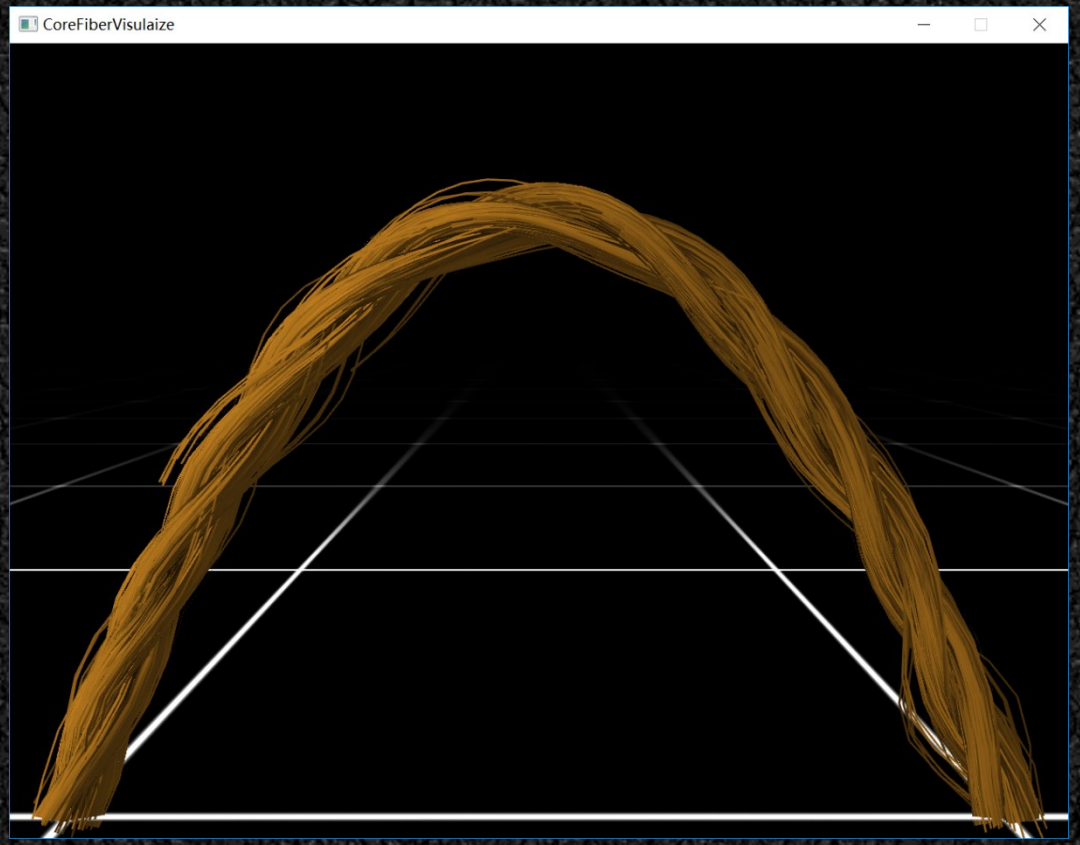
常规纤维+自由纤维

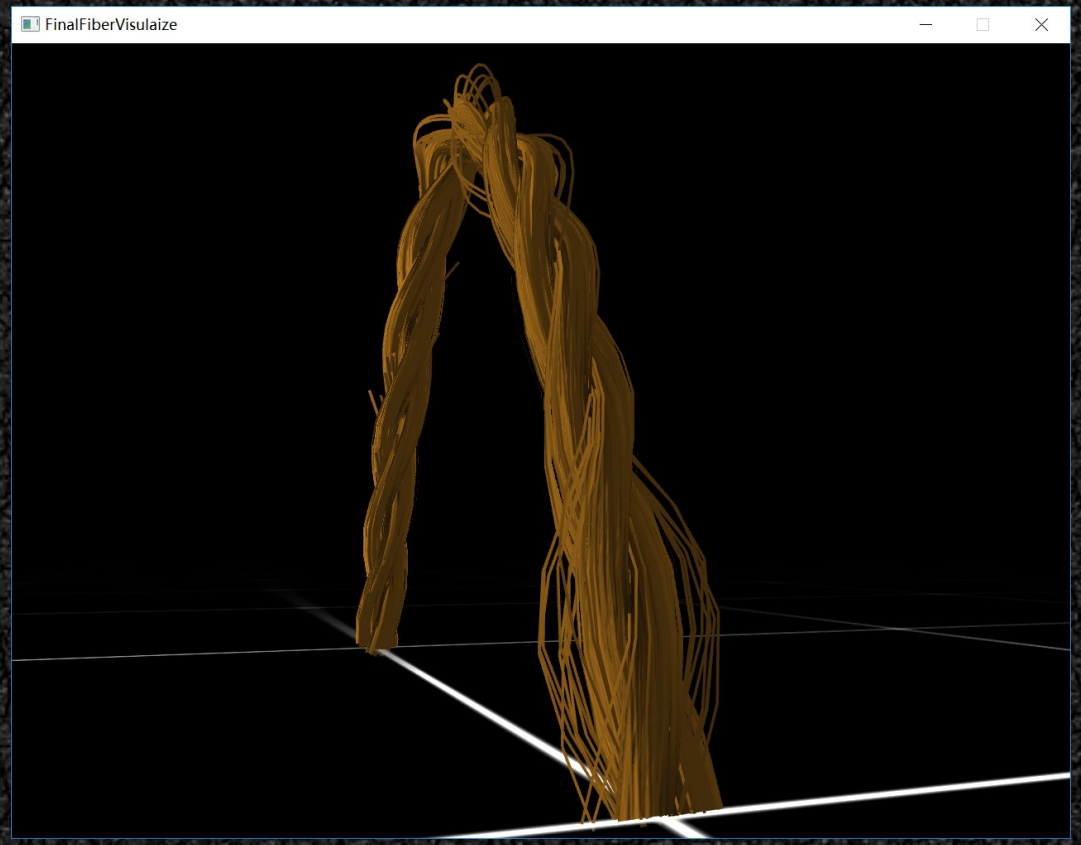
- 对非核心纤维中的三种类型纤维组合如下：



常规纤维+环形纤维+自由纤维

在GPU中对纱线几何模型进行生成：核心纤维+非核心纤维





- 细分纤维的数量有上限，核心纤维面片则弥补了这个不足，使得纱线看起来丰富。

在GPU中对纱线几何模型进行生成：柏林噪声

- 柏林噪声用于程序生成随机内容，可以用于生成波形，起伏不平的材质或者纹理。使用一维柏林噪声对自由纤维进行扰动如下：

- 纤维末端抖动的效果是可接受的，但是感觉自由纤维缠绕周期太长了，进而对缠绕周期从一个周期缩短为半个周期，剩下半个周期生成常规纤维。效果如下：

在GPU中对纱线几何模型进行生成：纤维位置的生成

对于纤维的初始半径与角度，师兄直接从fiber.txt中对初始横街面的纤维所在位置进行提取。然而fiber.txt并不是康奈尔大学经过对纤维CT进行扫描后得到的原始数据，其本身已经是按照模型匹配参数生成的文件。康奈尔大学是通过结合给定概率分布的rejection sampling来对纤维位置的分布进行采样的，故在对纤维初始化时采用同样的实现。概率分布函数如下所示：

$$p(R) = (1 - 2\epsilon)\left(\frac{e - e^{R}}{e - 1}\right)^{\beta} + \epsilon$$

在GPU中对纱线几何模型进行生成：纤维位置的生成

- 拒绝采样分为两个步骤：

① 在单位圆盘中进行均匀采样

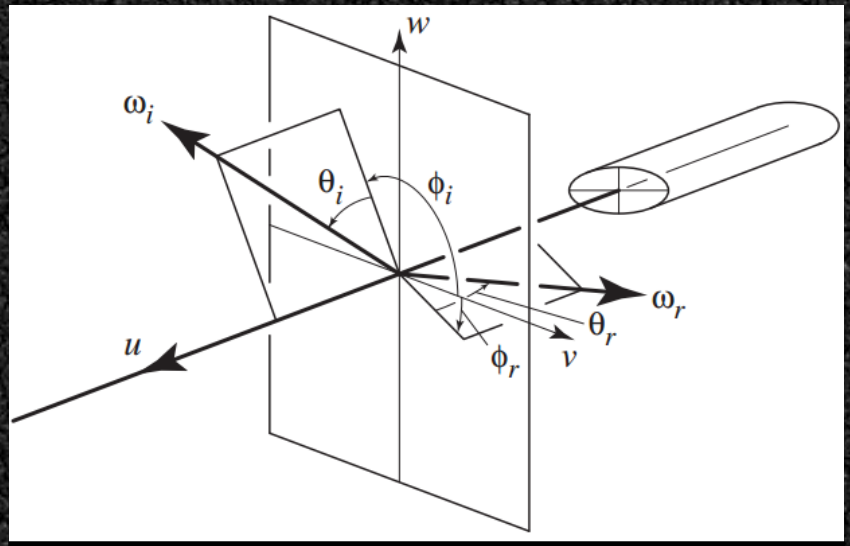
② 计算采样纤维点与单纱中心，即原点的距离，并根据函数求解相应的概率密度P，并与[0, 1]上的均匀采样点进行比较，若小于P，则接受样本，并对样本的初始半径与角度进行存储。需要注意的是初始角度需要从[0, 2PI]映射到[-PI, PI]。

Algorithm 2 Sampling cross-sectional fiber location

Require: cross-sectional fiber distribution parameters ϵ, β

procedure SAMPLEFIBERLOCATION(ϵ, β)
  repeat
    draw (x, y) uniformly in a unit disc
    draw ξ from $U[0, 1]$
  until $\xi < p\left(\|(x, y)\|_2\right)$ ▷ p defined in Eq. (1)
  return (x, y)
end procedure

在GPU中对纱线几何模型进行生成：着色模型

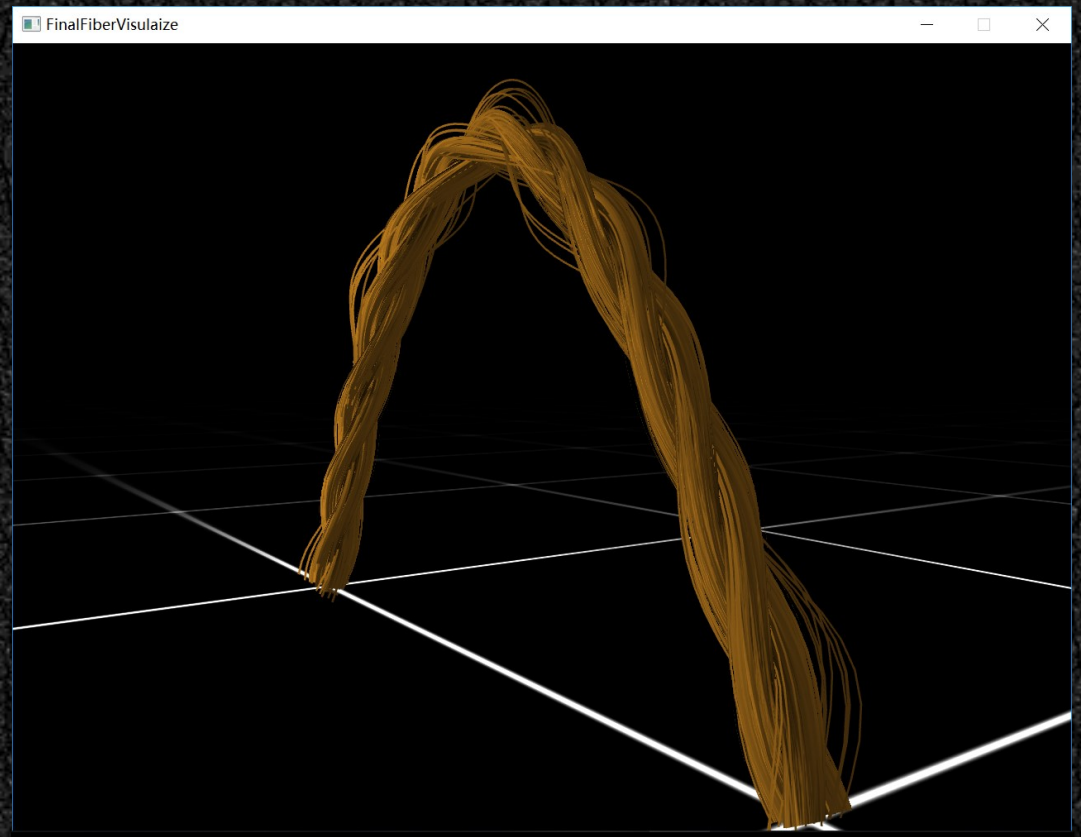


柱状体散射模型的几何符号表示

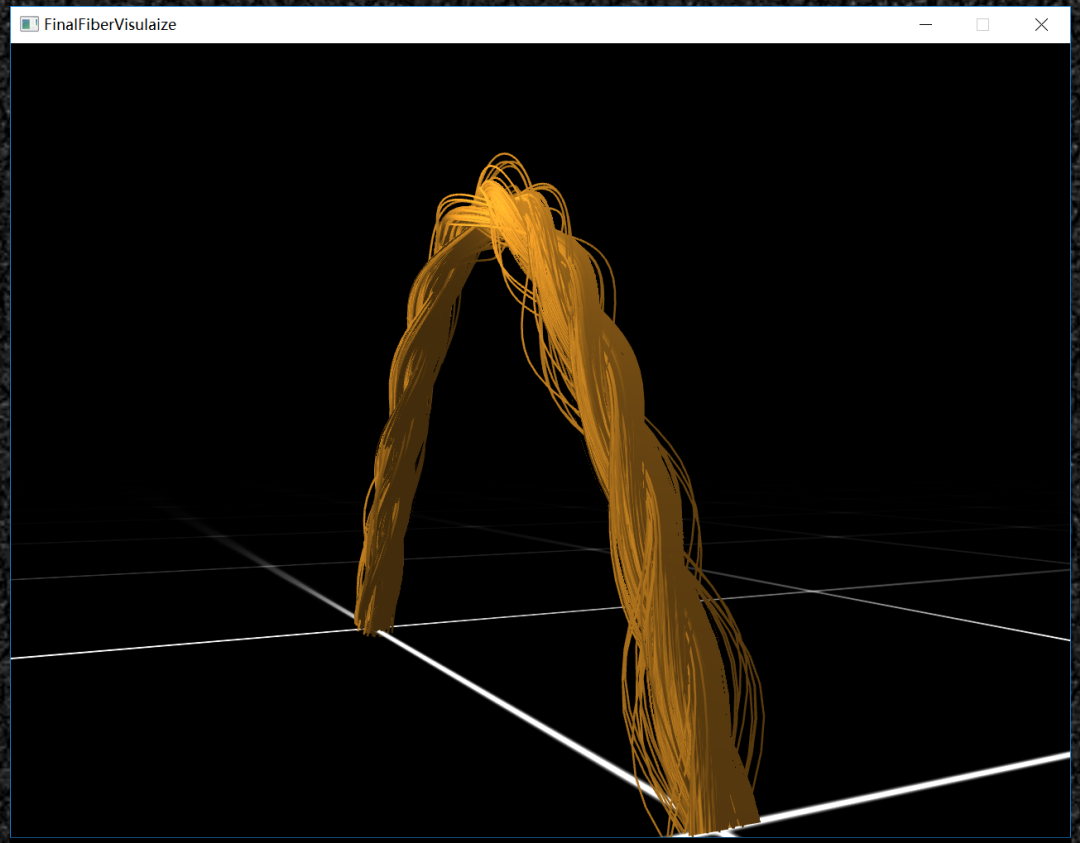
- 纤维的光散射模型同样由环境光、漫反射光、镜面反射光三部分组成，其中环境光和漫反射光都与Phong光照模型相一致，但镜面反射采用了柱状物体的散射模型。计算如下：

$$f_{\text{specular}} = k_{\text{specular}} \cos(\phi/2) \exp(-\theta_h^2/2\beta^2)$$

在GPU中对纱线几何模型进行生成：着色模型



Phong光照模型

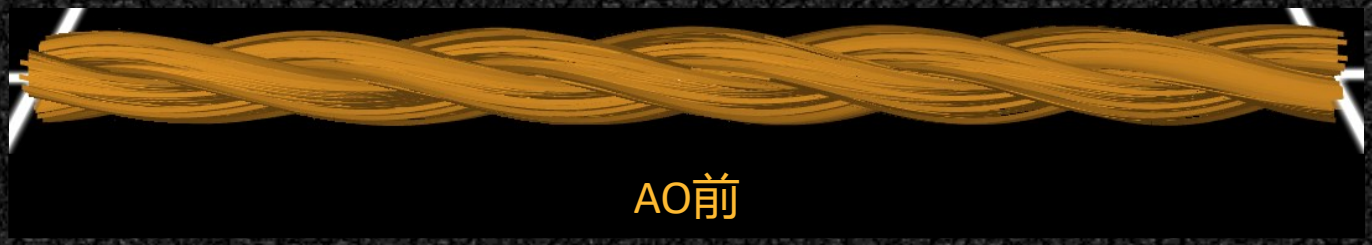


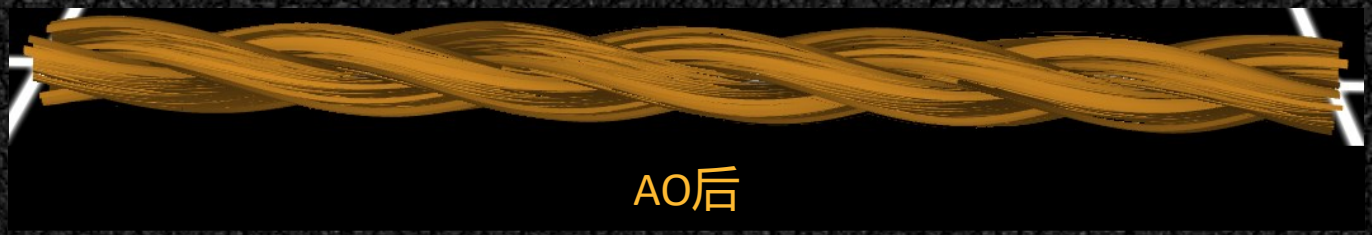
论文着色模型

在GPU中对纱线几何模型进行生成：基于距离的简单AO

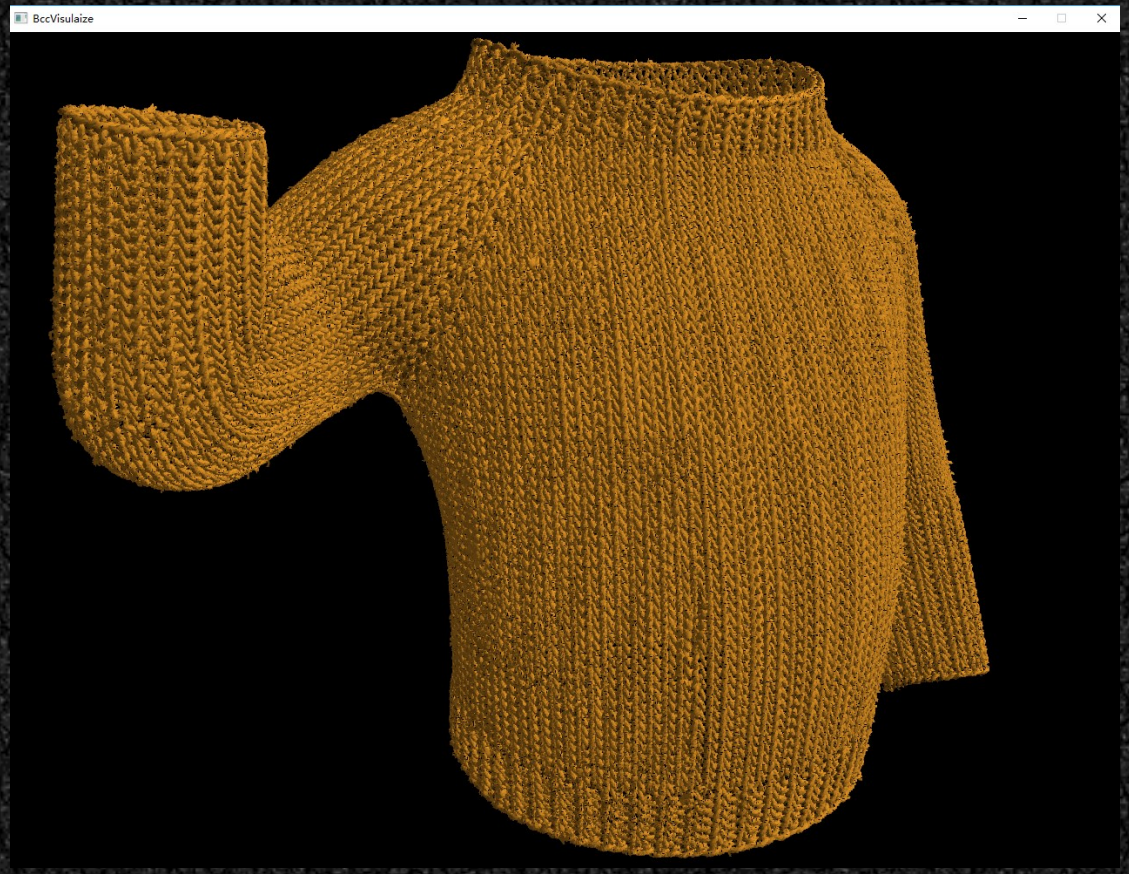
一种间接光照的模拟叫做环境光遮蔽(Ambient Occlusion)，它的原理是通过将褶皱、孔洞和非常靠近的墙面变暗的方法近似模拟出间接光照。这些区域很大程度上是被周围的几何体遮蔽的，光线会很难流失，所以这些地方看起来会更暗一些。

对于纱线，论文采用了一种简单的模型来对环境光遮蔽进行近似：依据与单纱中心的距离来对环境光进行缩放。使用AO前后对比如下：





在GPU中对纱线几何模型进行生成：完整织物模型



在GPU中对纱线几何模型进行生成：LOD，层次细节优化

- 纱线随着视点的远离会变得越来越细，这使得可以通过单纱与纤维的直径所占的屏幕像素宽度来对单纱所细分出来的纤维数量进行调整。给定阈值ω_{LOD}，论文对LOD缩放系数求解如下：

$$\lambda = \begin{cases} 1, & \text{if } \omega^{\text{LoD}} \leq \omega^{\text{fiber}} \\ 0, & \text{if } \omega^{\text{ply}} \leq \omega^{\text{LoD}} \\ \frac{\omega^{\text{fiber}}}{\omega^{\text{LoD}}} \left(\frac{\omega^{\text{ply}} - \omega^{\text{LoD}}}{\omega^{\text{ply}} - \omega^{\text{fiber}}} \right), & \text{otherwise,} \end{cases}$$

- 令N_{fiber}为最大可细分纤维数量，在细分控制着色器中对实际纤维数量调整如下：

$$n = n_{fiber}^{\max} * \lambda^2$$

在GPU中对纱线几何模型进行生成：LOD，层次细节优化

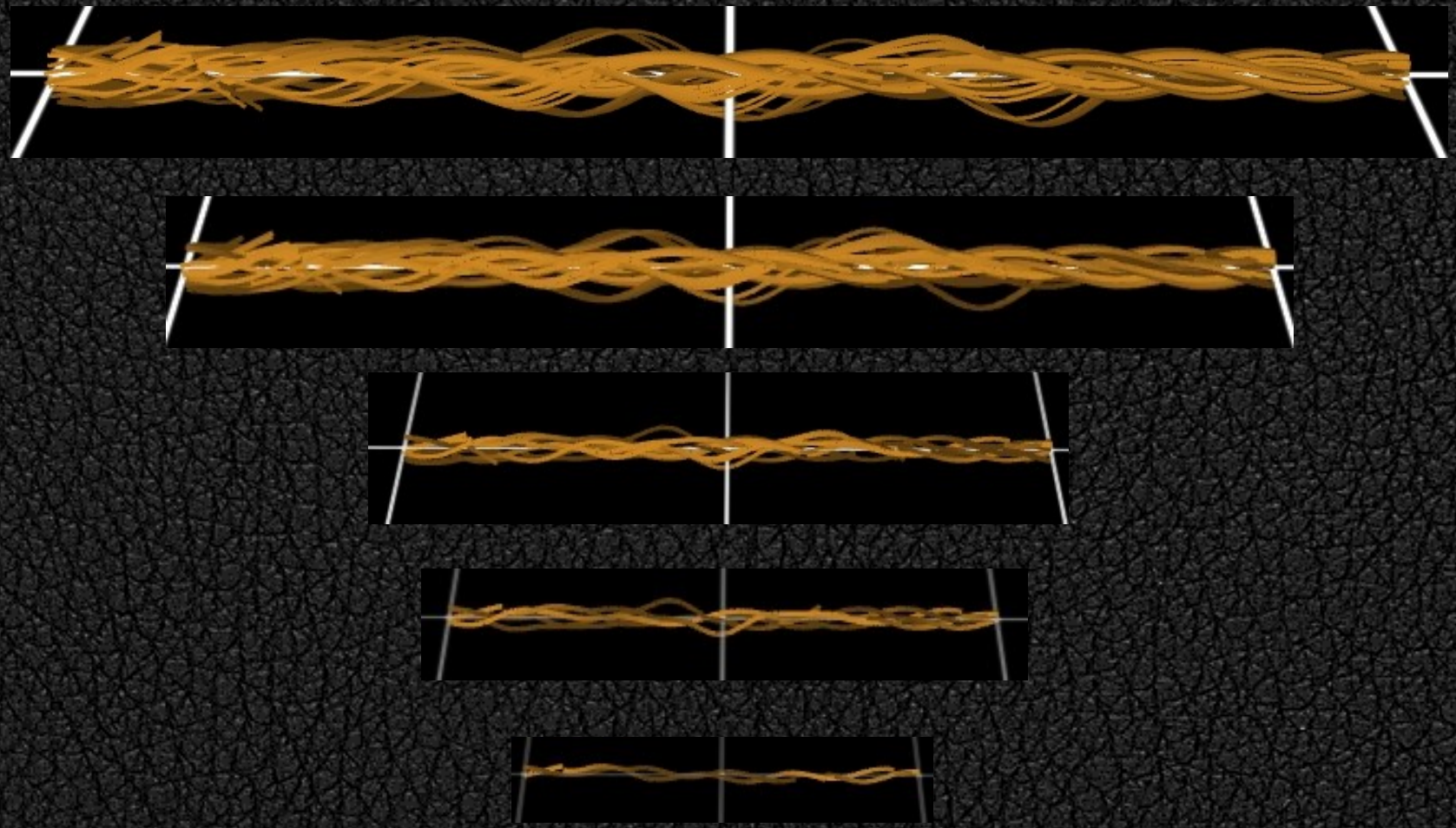
- 关于单纱直径屏幕像素宽度的求解，可以将单纱中心与纱线中心对齐，然后以视点连线方向为法向量对直径的两个端点进行扩展，再从世界坐标转换到屏幕像素坐标。纤维的像素宽度可以通过比例常量进行计算：

$$\frac{\omega_{fiber}}{\omega_{ply}} = \frac{R_{fiber}}{R_{ply}}$$

- 因为在GPU的裁剪之前进行了变换，所以还可以提前判断顶点是否位于投影面之外，对两端都位于屏幕之外的纱线不进行纤维的细分。需要说明的是由于纱线段很短，所以将两端位于屏幕外但穿过屏幕的情况进行相同的处理。

在GPU中对纱线几何模型进行生成：LOD，层次细节优化

近



远

在GPU中对纱线几何模型进行生成：LOD，层次细节优化

近

远

在GPU中对纱线几何模型进行生成：核心纤维的动态补偿

- 随着视点的远离，纤维数量减少，织物会显得单薄，故使用前面的λ^2对核心纤维的半径进行补偿，使得远处观察织物依然丰富：

补偿前

补偿后

在GPU中对纱线几何模型进行生成：将两遍渲染进行合并

- 前面的结果均由核心纤维贴图与非核心纤维两次渲染生成。两者都以纱线控制点作为输入，分开渲染是因为核心纤维贴图需要在细分着色器之后使用几何着色器将等值线线扩展成四边形面片，而非核心纤维只需要细分着色器参与。这样实现的好处是逻辑清晰，然而两次渲染中存在大量可以复用的数据，且织物的控制点数量庞大，绘制调用过多会很快达到性能瓶颈。以下列毛衣为例，共有146条曲线，14W+的顶点：

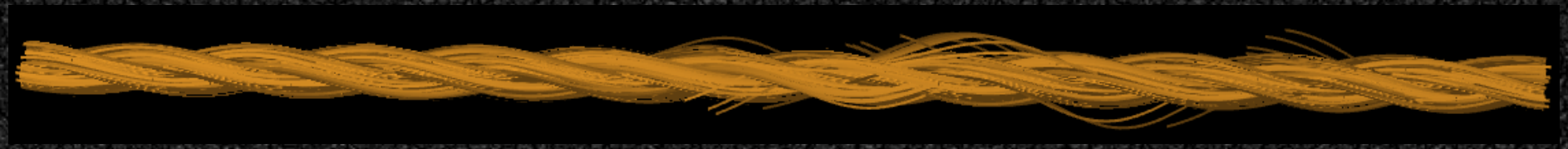
Alien Sweater Cables

Download:	**alien_sweater_cables.bcc**
Curve Type:	Catmull-Rom (uniform)
File size:	1.6 MB
Curve count:	146
Control points:	142,654

*Yarn-level model by Cem Yuksel. Cloth surface by **Rune Spaans**. Alien character model by **Christer Sveen**.*

在GPU中对纱线几何模型进行生成：将两遍渲染进行合并

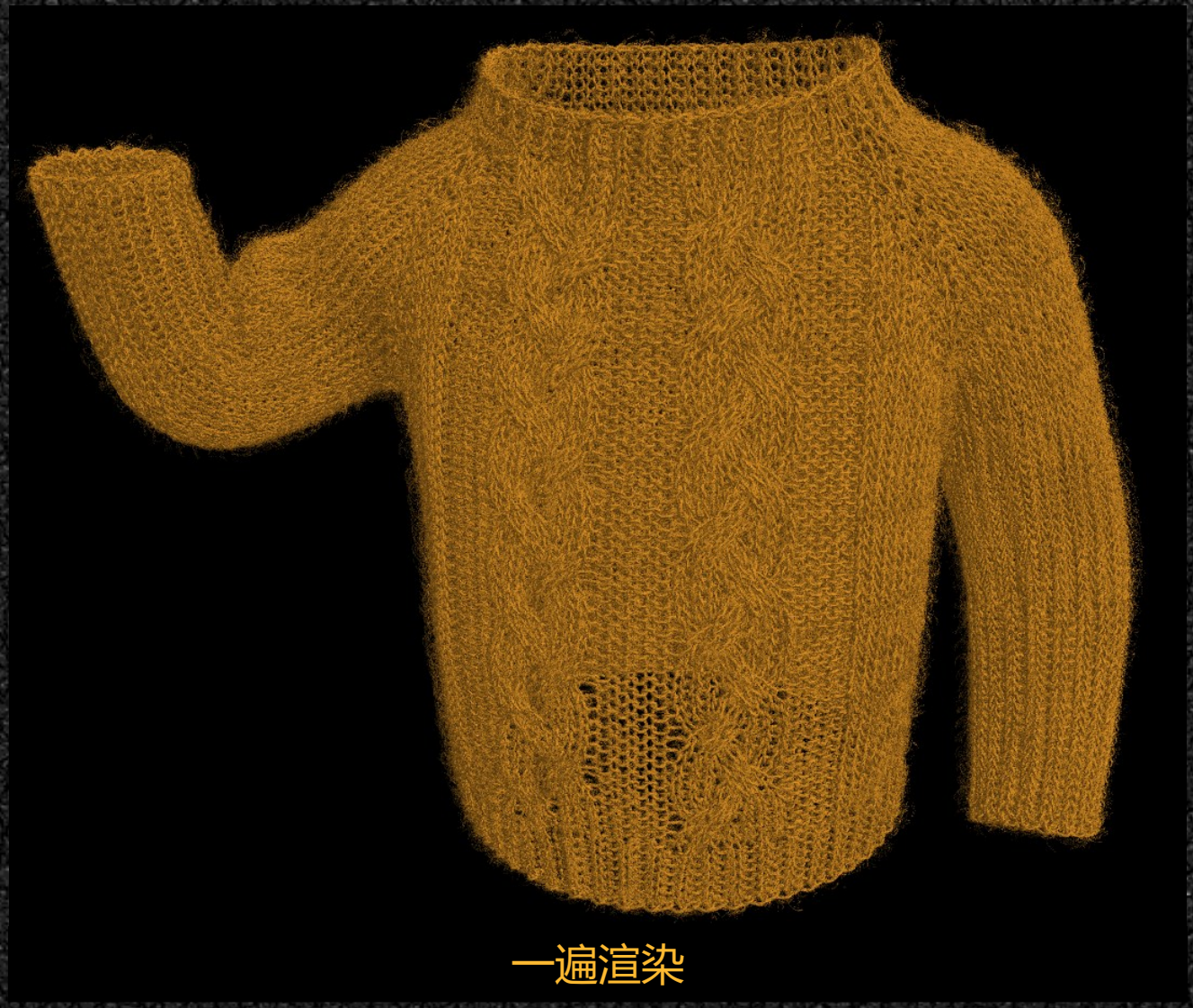
思路：一根纱线控制点细分为63根纤维，每条单纱平均21根，将第1条扩展成核心纤维面片，其余的20条为非核心纤维。但是由于都需要经过几何着色器，输入的线段不能再以线段的形式进行输出，故也以一个很小的宽度对非核心纤维进行扩展：



在GPU中对纱线几何模型进行生成：将两遍渲染进行合并

两遍渲染

在GPU中对纱线几何模型进行生成：将两遍渲染进行合并

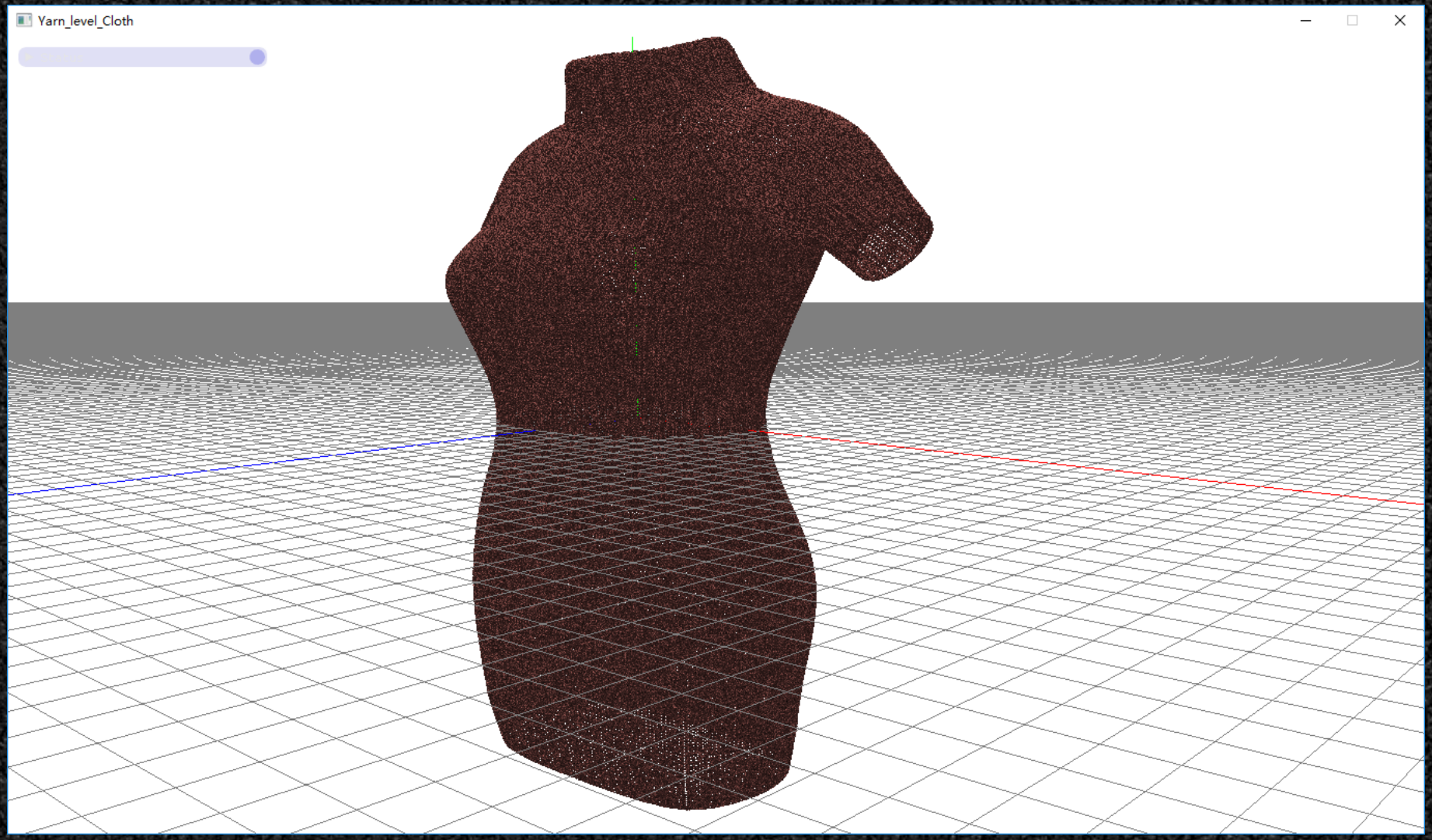


一遍渲染

毛绒感极大的增强

结果展示

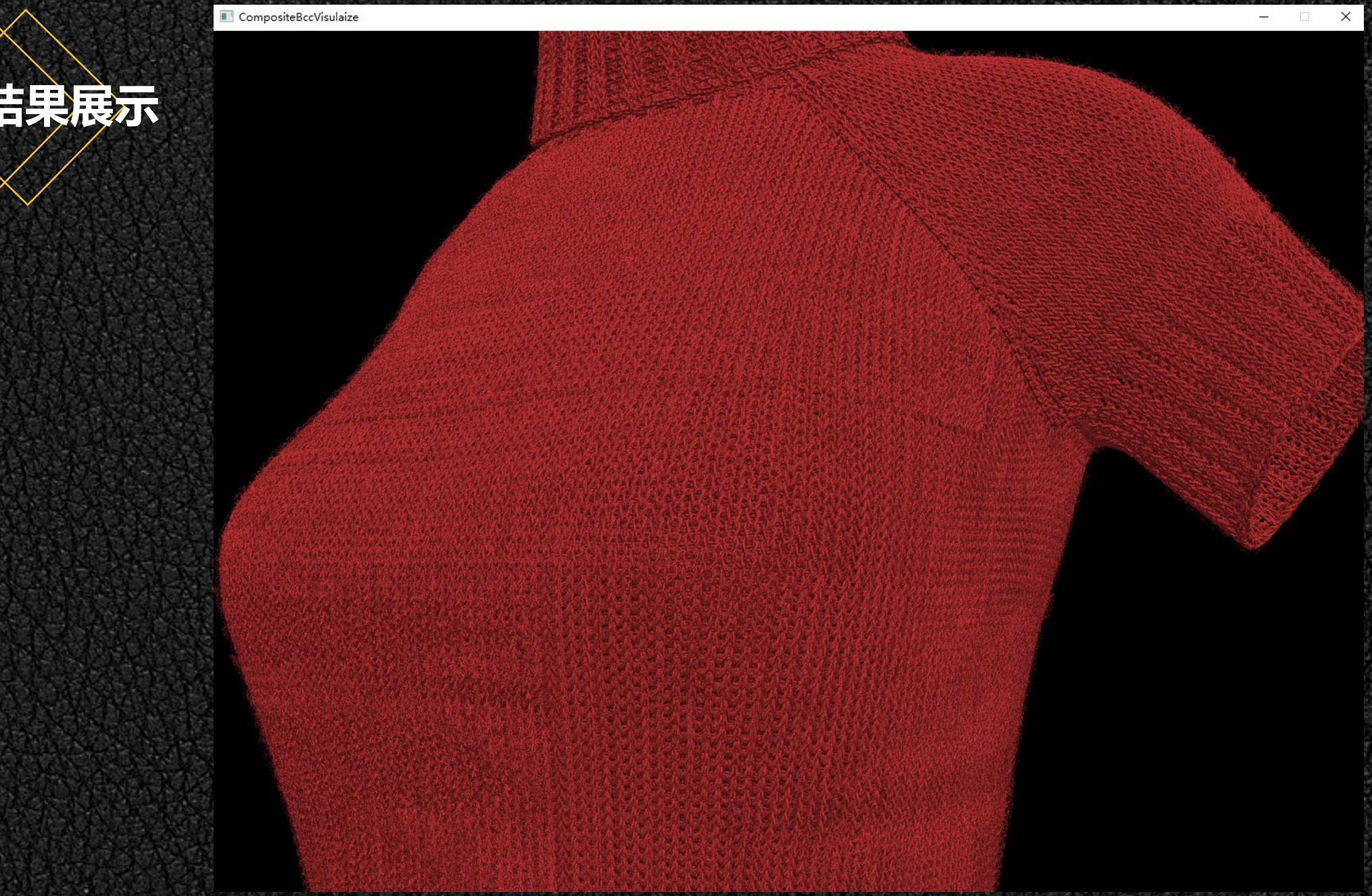
结果展示



结果展示

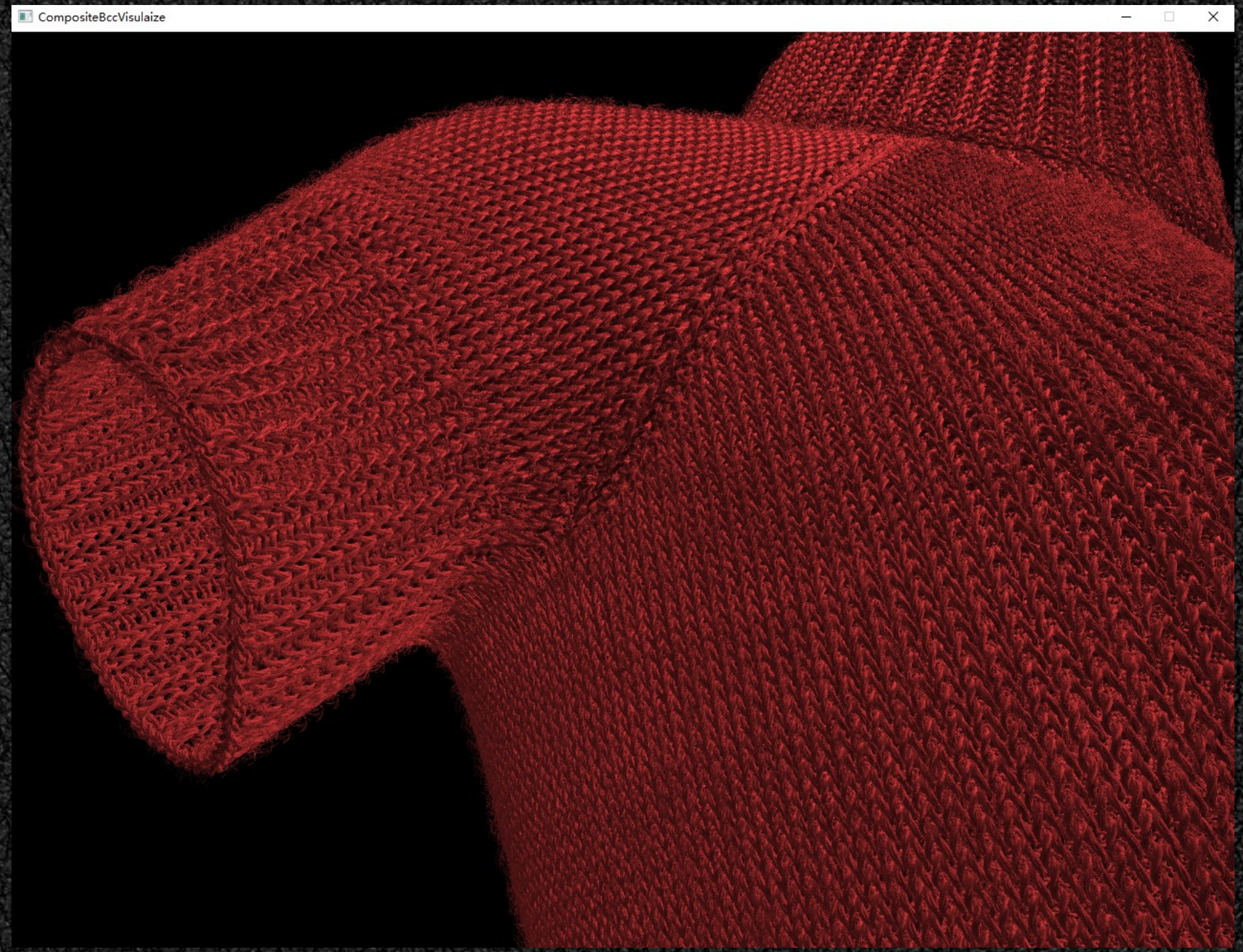
结果展示

结果展示

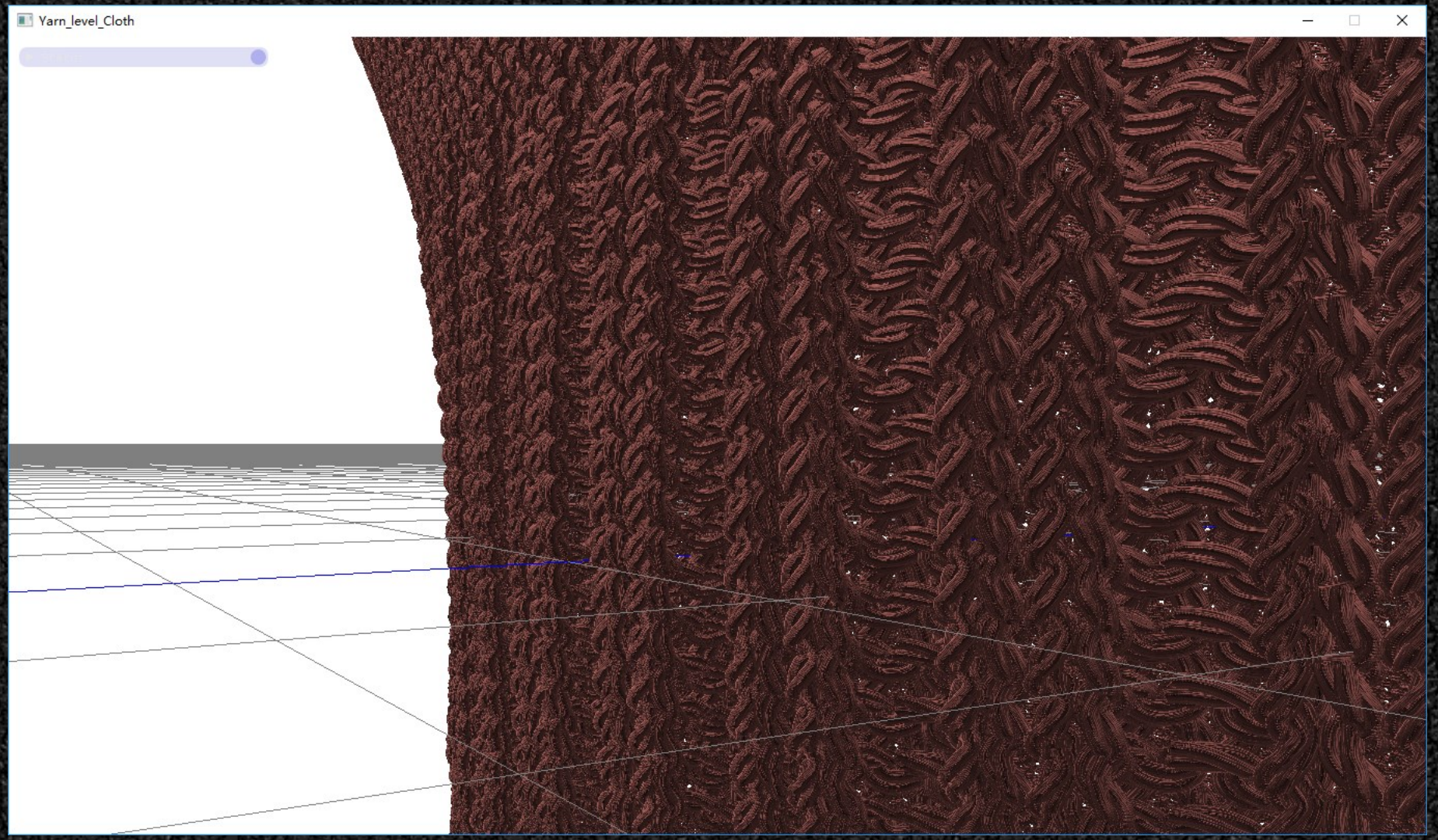


结果展示

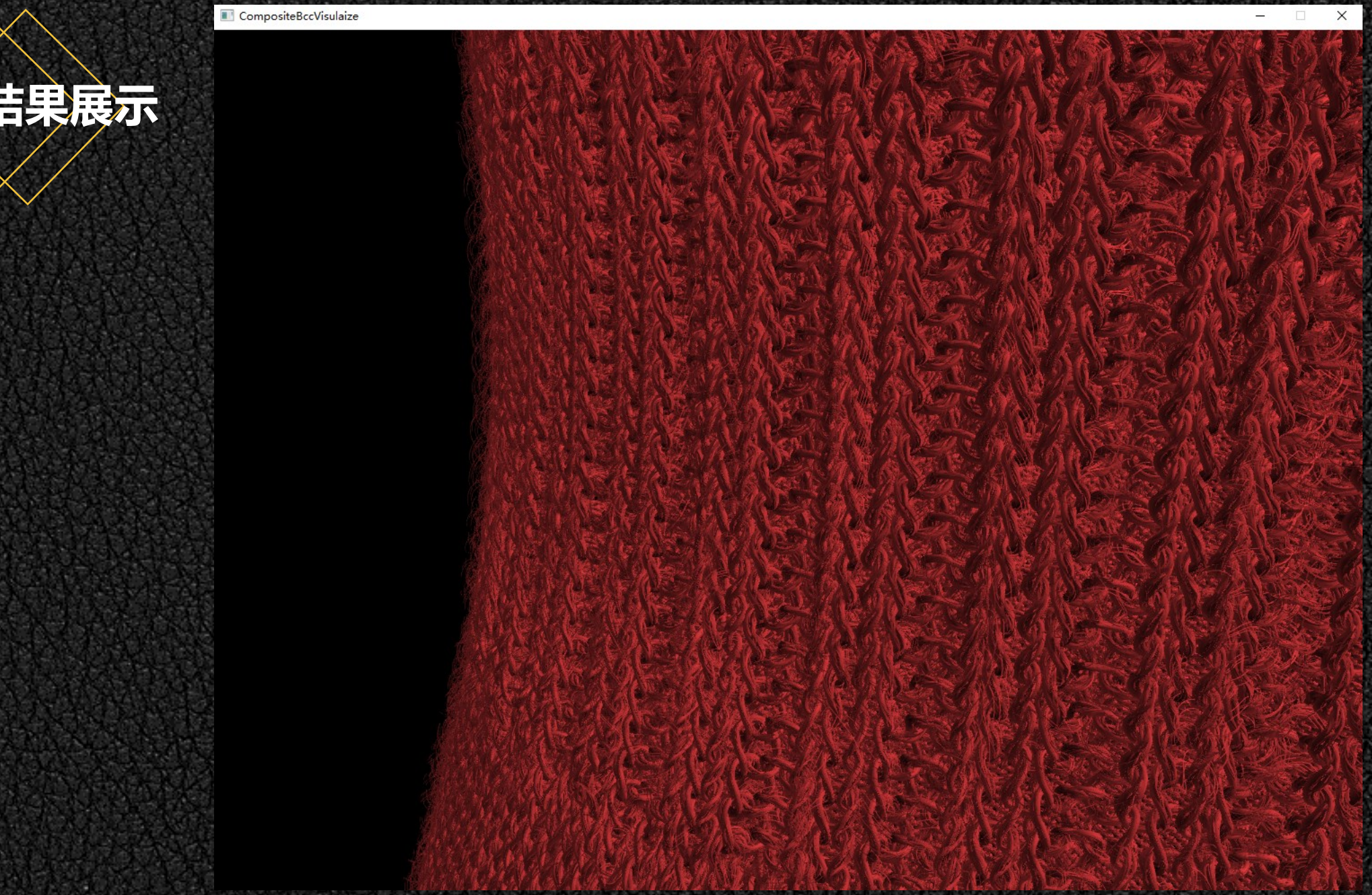
结果展示



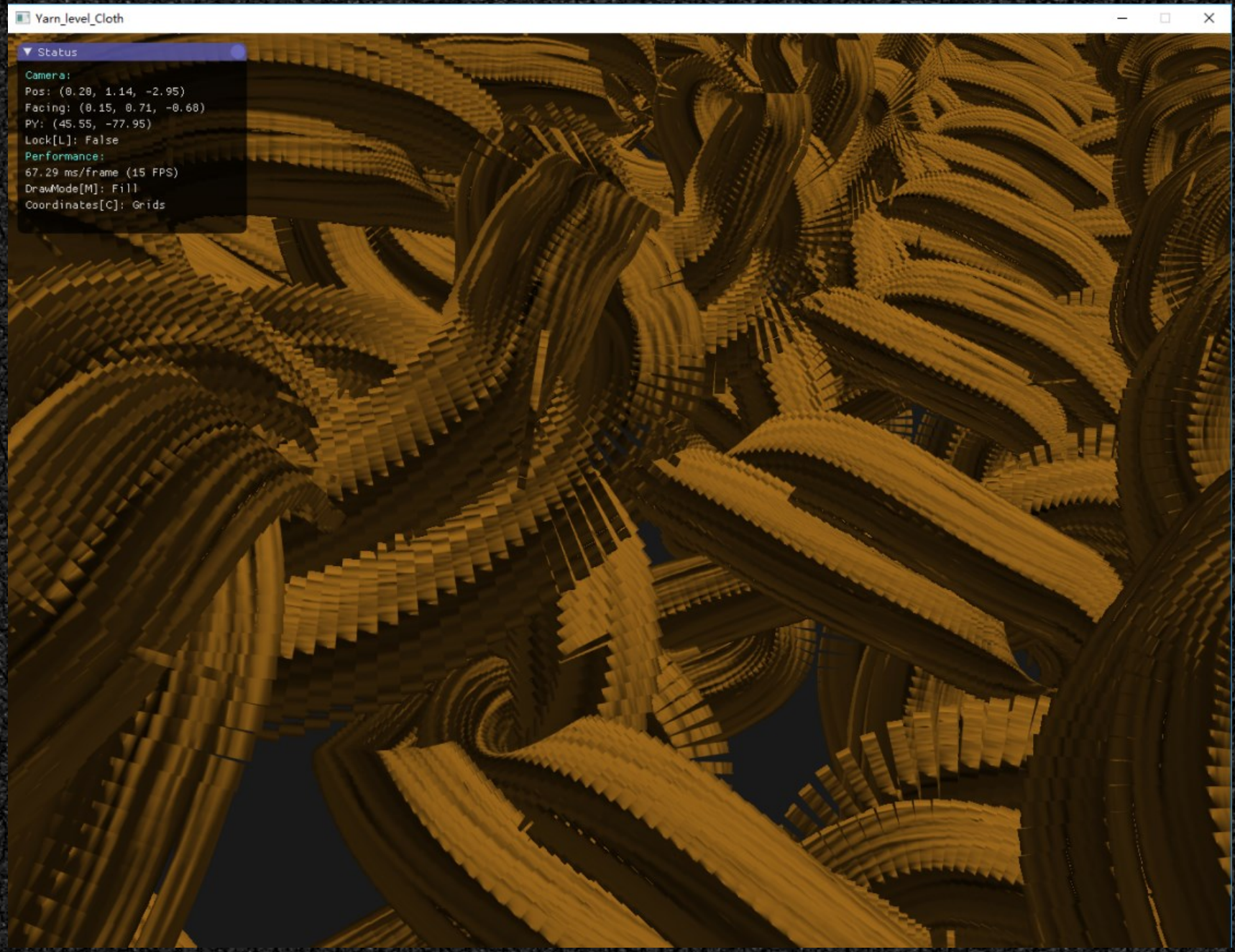
结果展示



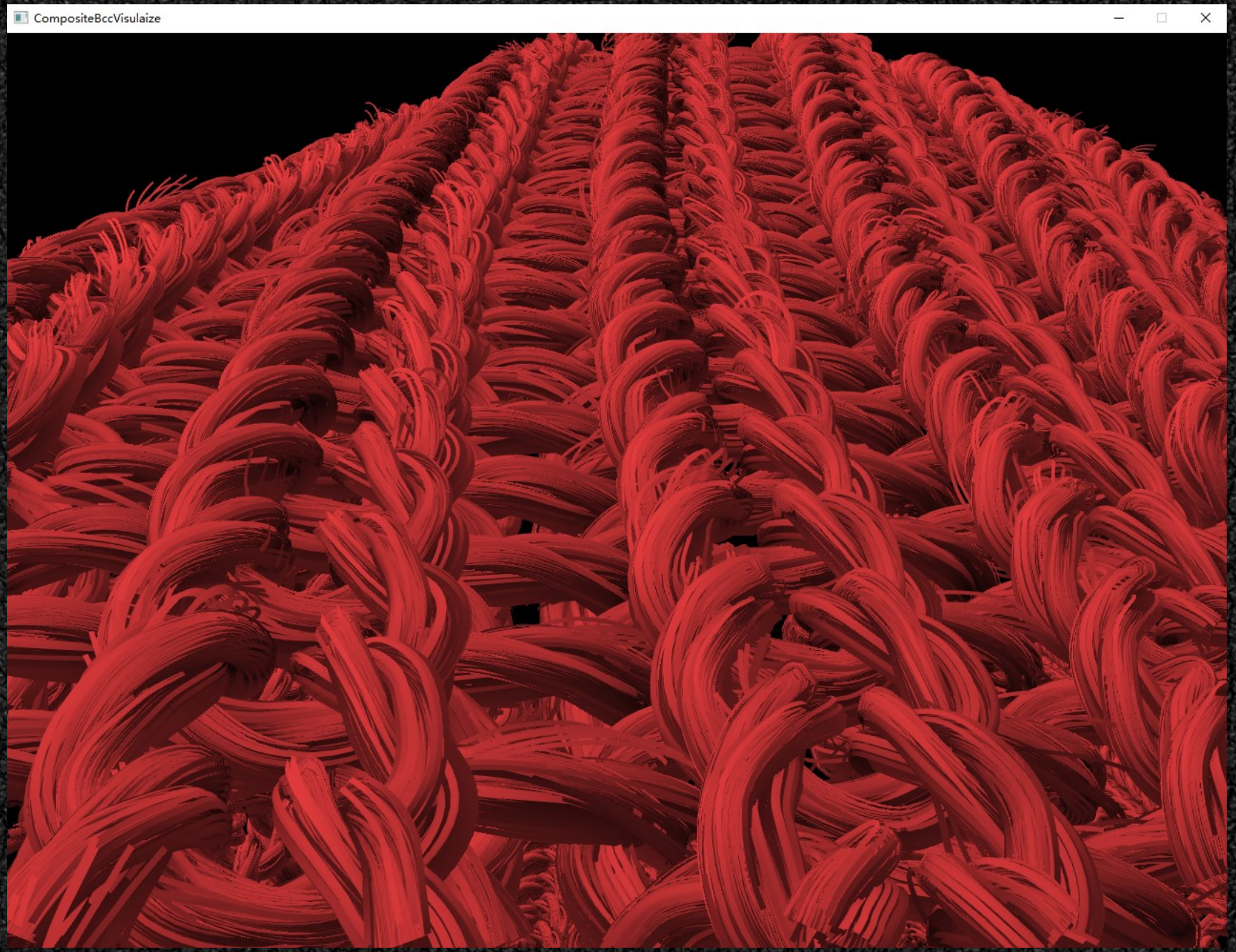
结果展示



结果展示

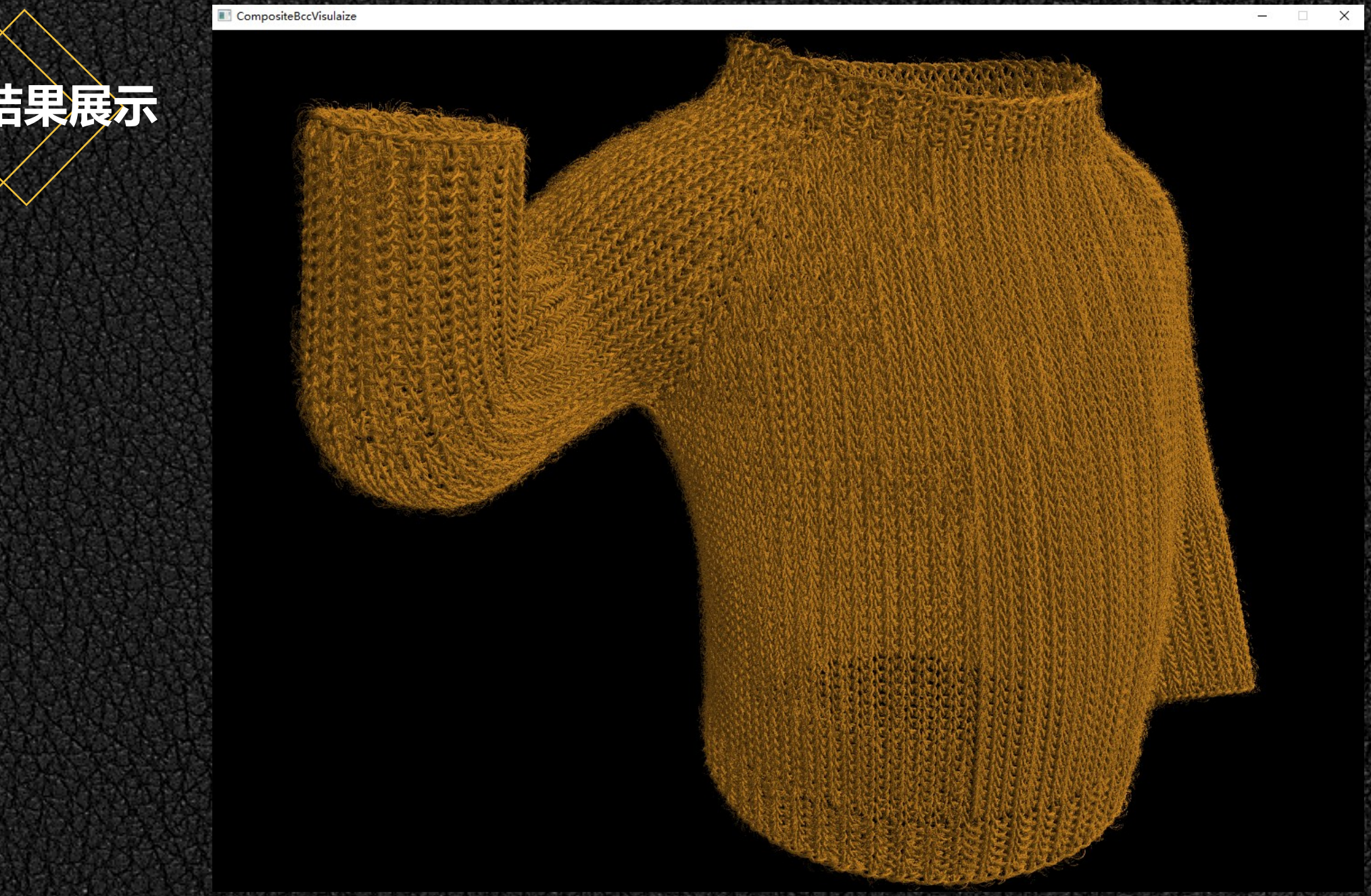

Yarn_level_Cloth
Status
Camera:
Pos: (0.20, 1.14, -2.95)
Facing: (0.15, 0.71, -0.68)
PY: (45.55, -77.95)
Lock[L]: False
Performance:
67.29 ms/frame (15 FPS)
DrawMode[M]: Fill
Coordinates[C]: Grids

结果展示

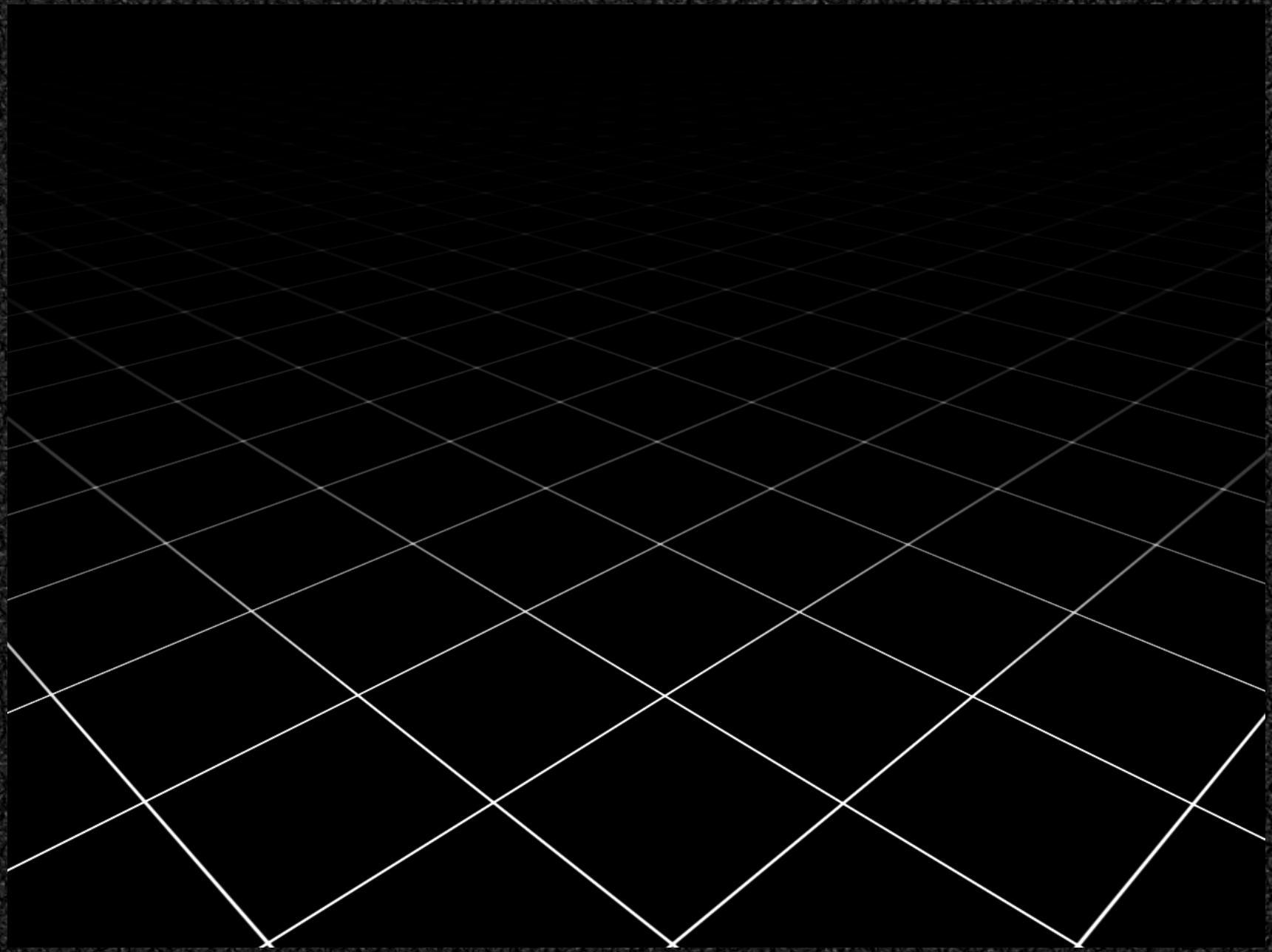


结果展示

结果展示



结果展示



谢谢观看