

Final project: 用两个三角形渲染世界

张高阳 | blurgy@gmail.com

2020 年 5 月 11 日

简介

- ▶ 选题: 2.4.2: 用两个三角形渲染世界
- ▶ 冰淇淋 emoji



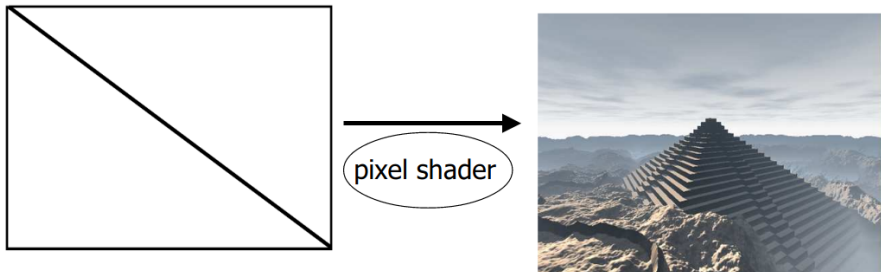
效果



<https://blurgy.xyz/cg/demo.mp4>

基本思想

- ▶ 用两个三角形覆盖整个屏幕
- ▶ 在 fragment shader 中构建距离场 (signed distance function, sdf)
- ▶ 使用 ray marching 方法逐像素渲染



图片来自 <https://www.iquilezles.org/www/material/nvscene2008/rwwtt.pdf>

目录

距离场

甜筒

奶油

糖果

着色

软阴影

材质

最后

最终结果

参考资料

距离场

甜筒

奶油

糖果

着色

软阴影

材质

最后

最终结果

参考资料

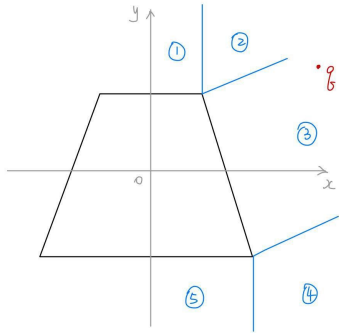
甜筒 - 分解

- ▶ 圆台 $\times 2$
- ▶ 碗形 $\times 1$
- ▶ 环 $\times 3$
- ▶ 竖线 $\times 7$



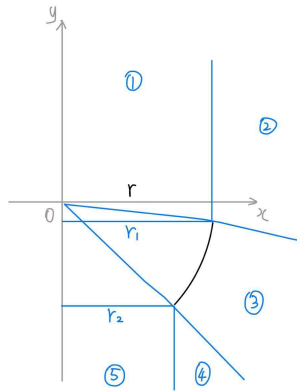
圆台

- ▶ 以 y 轴为中心对称
- ▶ `vec2 q = vec2(length(p.xz), p.y);`
- ▶ 梯形

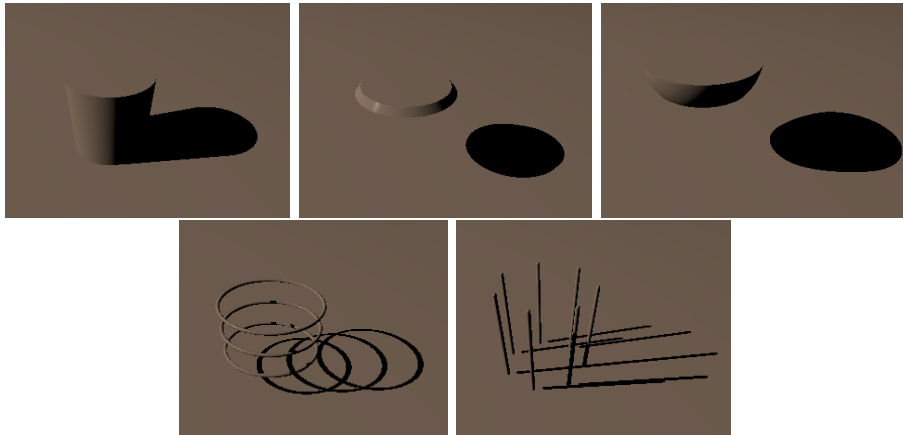


碗形

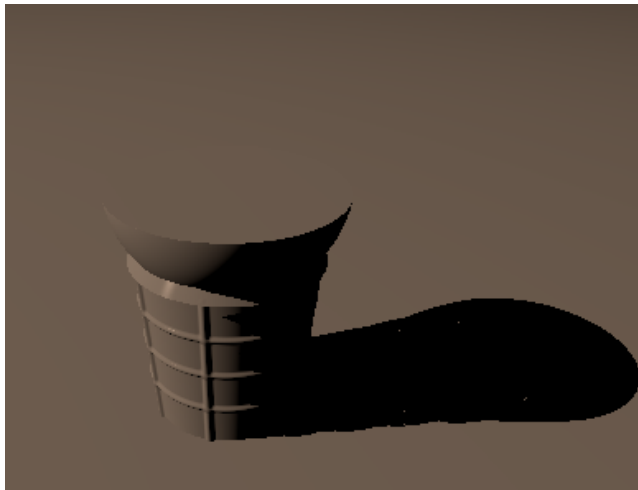
- ▶ $\text{vec2 } q = \text{vec2}(\text{length}(p.xz), p.y);$
- ▶ 圆 (部分)



五个部分



组合



`mapCone(p)`

奶油 - 分解

- ▶ 辐射状排列的锥形
- ▶ 分两层
- ▶ smooth blending



锥形

- ▶ 顶部半径为 0 的圆台

```
dist_cone = CappedCone(h, EPS, r);
```

- ▶ 平滑边缘:

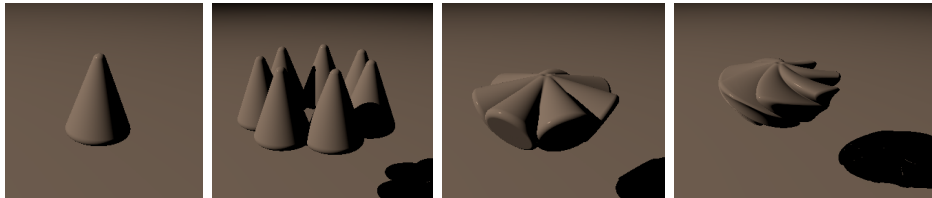
```
dist_cone = CappedCone(h-2*corner, EPS, r-corner) - corner;
```

重复和扭转

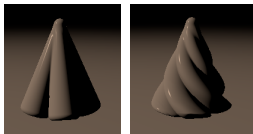
- ▶ 变换自身位置以求得不同的距离场
- ▶ `vec3 q = rotateY(p, TWOPI*i/rep);`
`dist = min(dist, Cone(q, h, r));`
- ▶ 根据 y 轴坐标做不同程度的扭转
`vec3 q = rotateY(p, -p.y * 6);`
`dist = min(dist, Cream(q));`

两层奶油

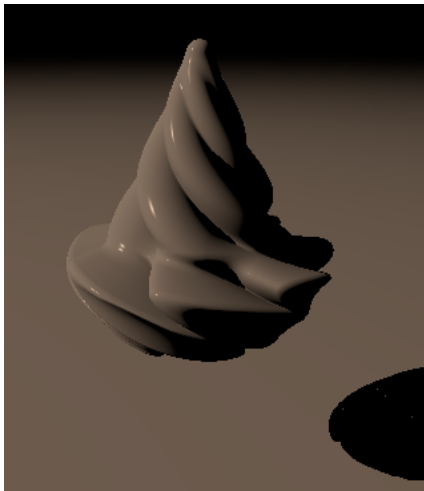
下层



上层



组合



`mapHead()`

糖果

- ▶ 立方体
- ▶ 圆角

旋转

- ▶ 随机初始方向

- ▶ 缓慢旋转

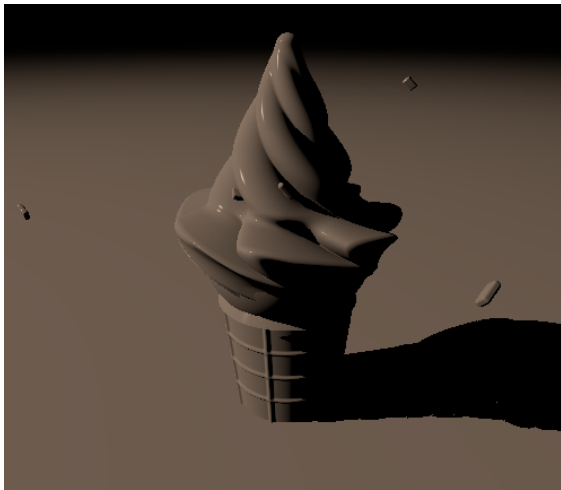
- ▶ `float delta_rot = 0.07*time + 0.05*sin(time);`
 - ▶ `vec3 q = inverse(rotateXYZ(ax+delta_rot, ay, az)) * (p-center);`
 - ▶ `dist = Box(q, box_shape);`

糖果



`mapCandies()`

组合



```
float dist = mapIceCream(p);
```

距离场
甜筒
奶油
糖果

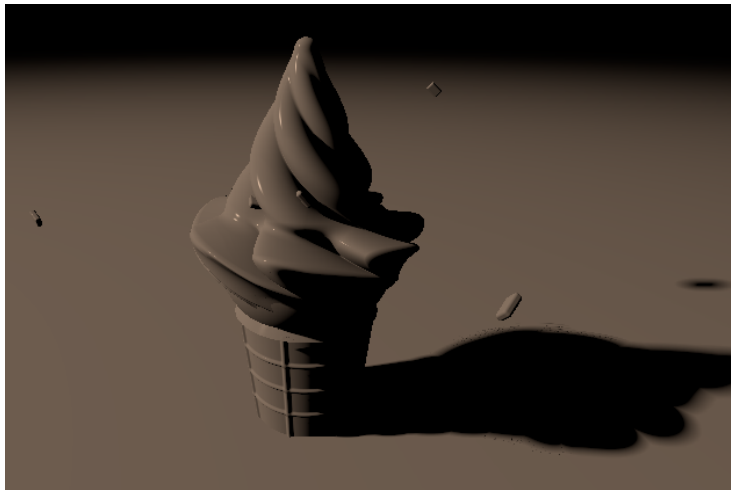
着色
软阴影
材质
最后
最终结果

参考资料

软阴影估计

- ▶ 在光线照亮点 p 的路径上:
 - ▶ 物体距离路径越近, 阴影越黑
 - ▶ 该物体离 ray marching 的起点越远, 阴影越浅
- ▶ 物体到起点的距离: t
- ▶ 物体到 ray marching 路径的距离: `float h = mapIceCream(ro + t*rd);`
- ▶ $k * h / t$

软阴影

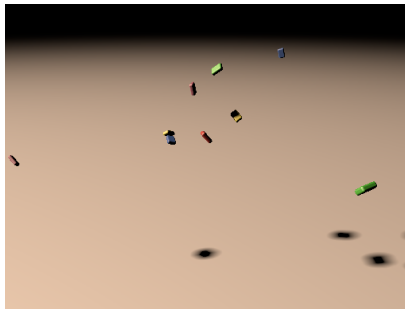
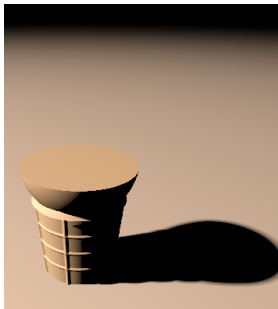


软阴影

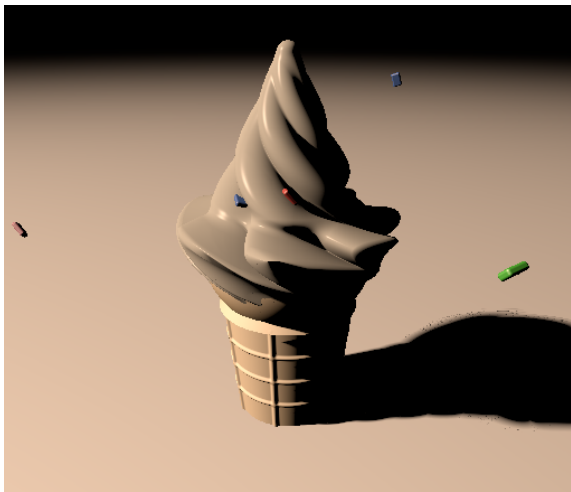
材质

- ▶ `Material{vec3 color; float roughness}`
- ▶ color: 漫反射颜色
 - ▶ `float diffuse = clamp(dot(n, l), 0, 1) * softShadow(p);`
- ▶ roughness: 控制高光
 - ▶ `float pn = exp2(10 * (1-roughness));`
 - ▶ `float specular = pow(clamp(dot(h, n), 0, 1), pn) * diffuse;`
- ▶ ray marching 过程中更新

添加材质



组合



最后

- ▶ 物体表面某点 p 可能因为周围的复杂形状而接收到较少环境光照
- ▶ 沿表面法向量向外取若干个点 q (5 个), 根据 q 点到世界的距离 $d=\text{map}(q)$ 和 q 点到 p 点的距离 h , 就可以估计出点 p 处的环境光遮蔽系数 occ .
- ▶ 反走样 (SSAA 2x)

最终结果



最终结果

距离场

甜筒

奶油

糖果

着色

软阴影

材质

最后

最终结果

参考资料

参考资料

- ▶ Vulkan API: <https://vulkan-tutorial.com/>
- ▶ ray marching 方法及技巧, 环境光遮蔽估计, 通过空间变换求得绕轴对称的若干 primitive 等:
 - ▶ <https://www.iquilezles.org/www/material/nvscene2008/rwttt.pdf> 第 48 页
 - ▶ <https://www.youtube.com/watch?v=8sCLZcvC00o>
- ▶ smooth blending: <https://iquilezles.org/www/articles/smin/smin.htm>
- ▶ 修正的圆台距离场函数: <https://iquilezles.org/www/articles/distfunctions/distfunctions.htm>
- ▶ 软阴影估计方式:
<https://iquilezles.org/www/articles/rmshadows/rmshadows.htm>

感谢观看!