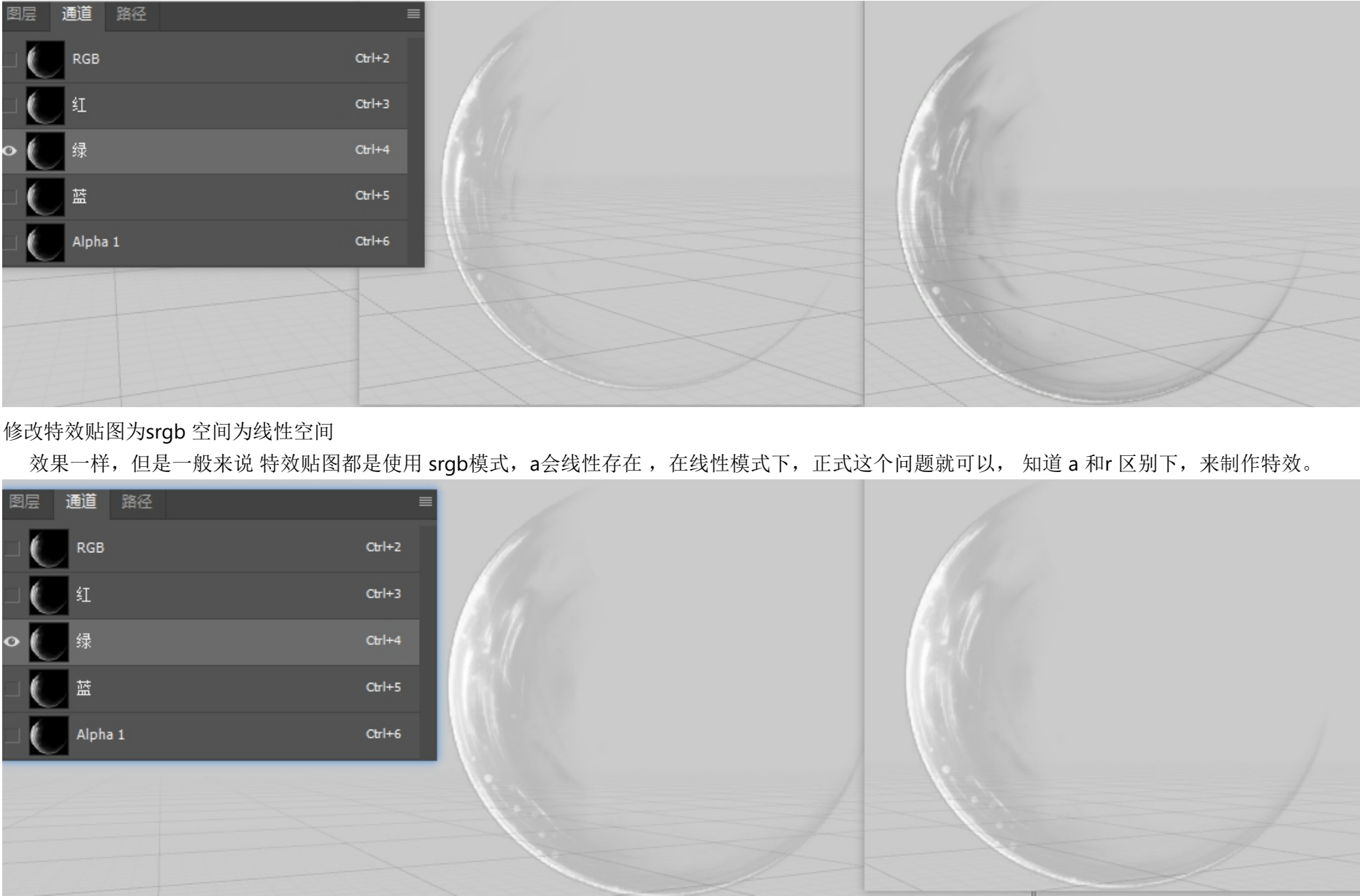


特效一些问题

线性模式

1.a 和r 通道问题

- 特效贴图 **rgba** 数据一样，但是实际特效显示效果不一样，原因是特效贴图的 **rgb** 通常会使用 **srgb**模式，而对于**a**通道还是使用线性模式。
 - 使用**r**作为透明进行特效渲染和使用**a**通道进行渲染
rgb 一般会人物是美术绘制图形，直接在伽马空间进行，而**a** 通道反应的是一个线性关系
第一效果是**r** 第二效果是**a** 由于使用**srgb** 会乘以伽马系数，导致透明的多
- 修改特效贴图为**srgb** 空间为线性空间
效果一样，但是一般来说 特效贴图都是使用 **srgb**模式，**a**会线性存在，在线性模式下，正式这个问题就可以，知道 **a** 和**r** 区别下，来制作特效。



blend模式下黑问题

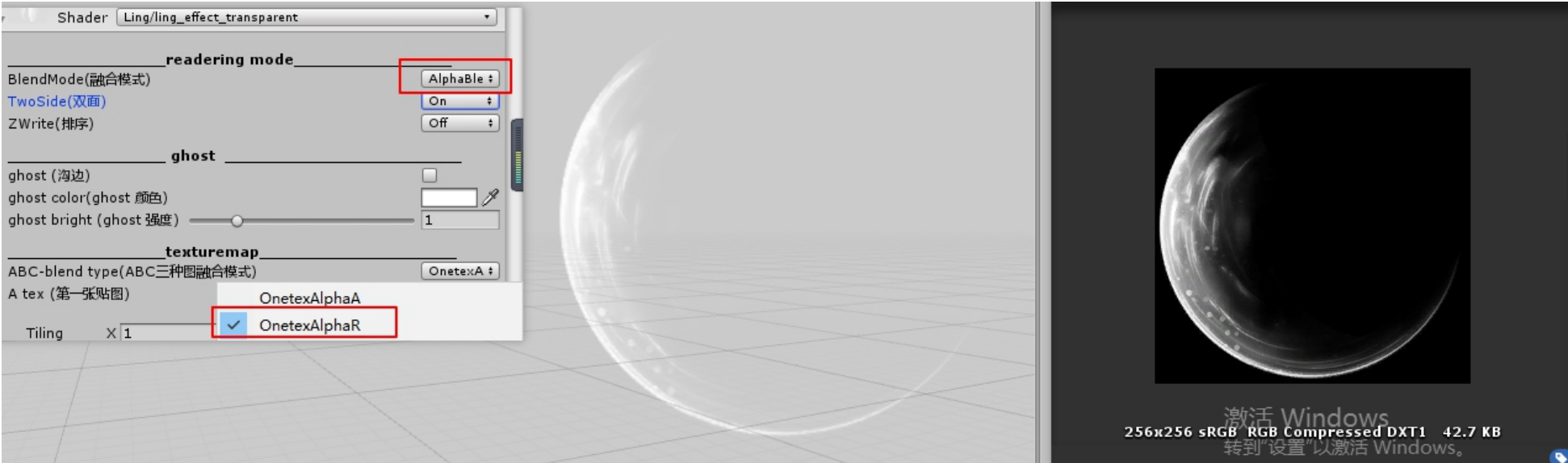
- 正常情况 使用blend 情况下 就是 $\text{srcAlpha} \cdot \text{OneMinusSrcAlpha}$ ，渲染的图形乘以自己的**a**通道 然后叠加上 背景 乘以**1-a**通道完成blend的操作。
- 对于黑白图，一般解决都是，**rgb**在原来亮度范围上加大加亮，使用小于一圈的**a**，这样在进行上面blend时候，渲染出来的图本身就是白色，在乘以**a**通道，来解决黑边问题。
测试操作，直接把**rgb** 变色了白色，blend出来没有黑边问题。



对于ling项目 要求大多数的特效图为去色模式，去掉**a**通道，，**a**通道采用自己的**r**通道代替，所有颜色颜色通过叠加进行，**rgb**本身采用**r**通道作为**a**通道进行blend时候，就会出现上面的黑边现象。
解决方法：在使用**r**通道作为**a**时候，**rgb shader** 直接使用1.0代替，解决blend 右黑边问题

```
#elif _ABCTEX_ONETEXALPHAR
float4 col = tex2D(_MainTex, ling_uv0);
//float3 rgb = col.rgb *ling_vertercolor.rgb *_MainColor.rgb *_MainBright;
float3 rgb = 1.0f ; |
float a = col.r *ling_vertercolor.a *_MainColor.a *_MainBright ;
result = float4(rgb , a );
#elif _ABCTEX_MULTWOTEXALPHAA
```

修改后 rgb图，blend模式，直接切换成 "OneTexAlpahaR" 黑边解决

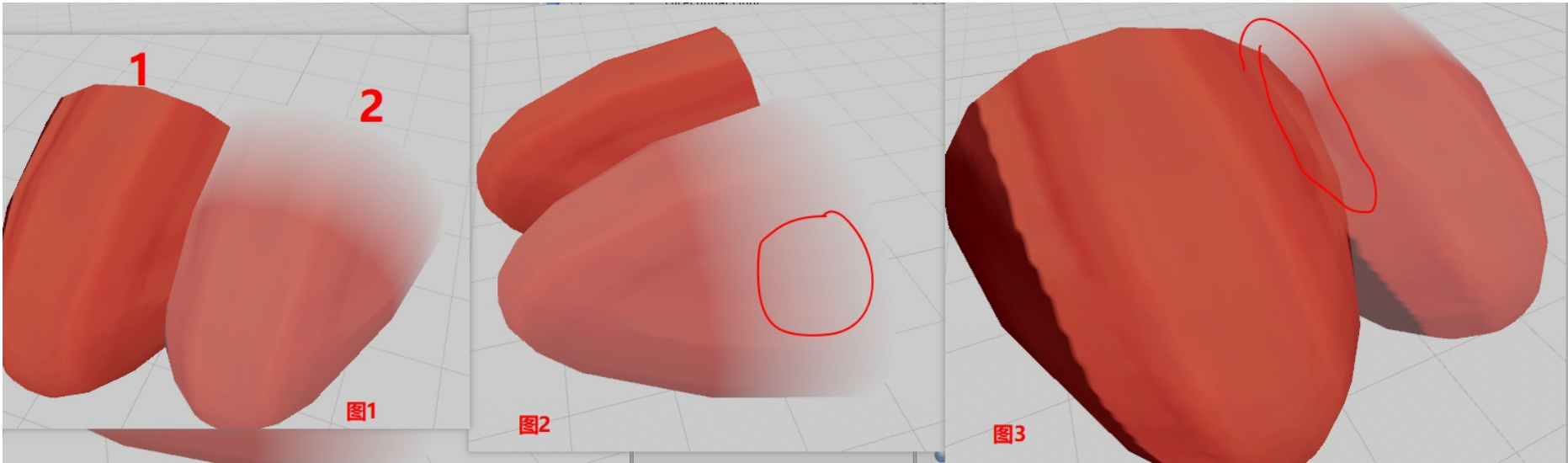


shader Geometry队列问题

- unity中给出的队列关系

渲染队列	渲染队列描述	渲染队列值
Background 【背景】	这个队列被最先渲染。它被用于skyboxes等。	1000
Geometry 【几何体】	这是默认的渲染队列。它被用于绝大多数对象。不透明几何体使用该队列。	2000
AlphaTest 【透明度测试】	用于alpha测试的几何体，它和Geometry队列不同，在所有几何体物体绘制后再进行渲染的对象，它更有效率。	2450
Transparent 【透明】	该渲染队列在Geometry和AlphaTest队列后被渲染，采用由后到前的次序。任何通过alpha通道混合的（也就是说，那些不写入深度缓存的Shaders）对象使用该队列，例如玻璃和粒子效果。	3000
Overlay 【覆盖】	该渲染队列是用于实现叠加效果的。任何最后被渲染的对象使用该队列，例如镜头光晕。	4000

- 如果使用Geomotgry 队列 shader 有z深度排序，融合模式就是 one zero
 - 使用 blend add 效果不对，逻辑也错误



1. 1模型 使用 Geomety 渲染队列 blend 模式 ONe zero
2. 2模型使用Geomety 渲染队列 blend 模式是 srcAlpha OneMinusSrcAlpha
3. 图1 渲染一个错误 blend模式 模型自身有了透明处理，但是和 后面 几何体 排列奇怪 并会出现 透明
4. 图2 也是说明 本身blend 模式，但是排序还是按照 Geomety 处理
5. 图3 角度出现乱序问题

- 2模型修改成 **Transparent** 模式，逻辑正确，上面问题都解决

