5.0 2D 镜头光晕

Yossarian King

镜头光晕是一种光学特效，它是当摄像机对向强光时，镜头的元件间相互反射而产生的。产生的最终效果就是从光源发散射来的一些透明的形状和颜色的变化图案。这种效果经常可以在电视中看到，当太阳进入到摄像机视野时就会出现。

在现实生活中，镜头光晕被认为是种缺陷，摄像机生产厂商通过使用特殊的镜头涂层来尽力避免光晕产生。不过视频游戏喜欢强调和夸大现实中所有让人觉得“酷”的地方，毫无疑问，镜头光晕就很酷。真实的镜头光晕靠光与摄像机光学系统的各个面复杂的交互作用产生。视频游戏中的镜头光晕只关注最终呈现的画面效果。本文讲述了一个迷人的镜头光晕特效的实现方法，此方法不用知道任何关于物理光学的知识，只需使用少量代码和美工。

5.0.1 方法

真实的镜头光晕产生于摄像机的光学系统中，所以自然呈现在所观看的景物“之上”。光晕的每个元素都是光从第二个镜片到前一个镜片上的反射。由于镜片是沿镜头中轴严格对齐的，因而在最终形成的画面上，这些反射成一条直线对齐，并且反射到画面中心的距离与相应的第二个镜片到前一个镜片的距离成比例。

基于以上观察研究，可以将镜头光晕作为 2D 问题来处理。光晕作为 3D 场景上的一个叠层来渲染，光晕元素沿着从光的投影位置到屏幕中心的直线渲染，如图 5.0.1 所示。

在此，我们不管所有物理光学方面的问题，而集中从美术效果上来考虑。镜头光晕特效采用一个小的纹理集来渲染，纹理集中的每一个元素为一种光晕元素类型——圆、圆环、六边形、辐射形等等，如图 5.0.2 所示。灰度纹理与顶点颜色结合以产生弱着色效果，使用 Alpha 混合产生半透明效果，元素以不同的大小渲染。

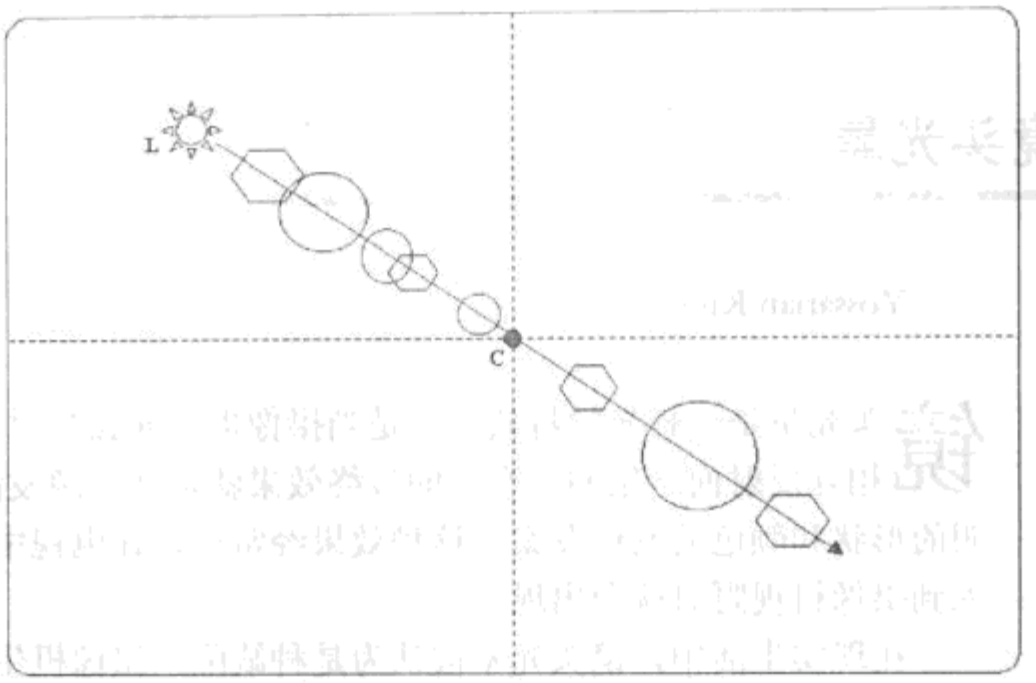

图 5.0.1 镜头光晕渲染是一个 2D 问题。镜头光晕的元素沿着从投影光到屏幕中心的直线渲染

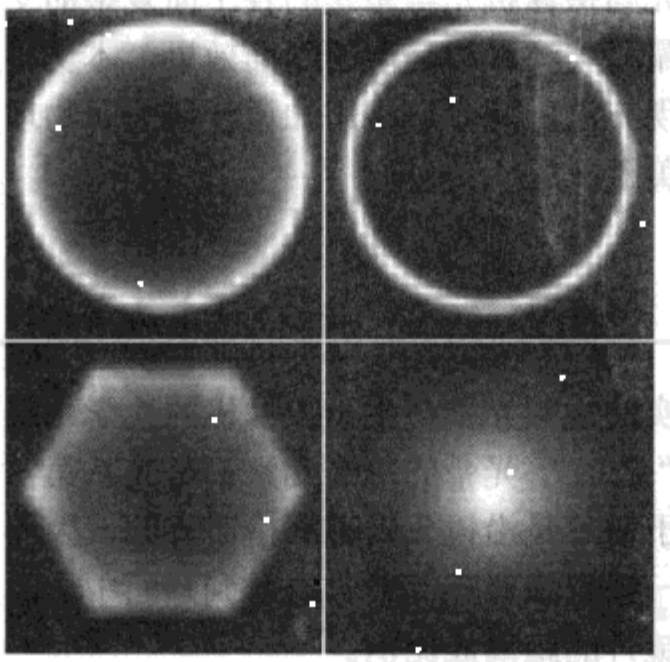

图 5.0.2 镜头光晕特效使用一个小型灰度纹理集来渲染

要达到真实的效果，镜头光晕还必须随着摄像机的移动产生相应的运动。总体上光晕的移动是由追踪从投射光位置到屏幕中心的直线来决定的。变化光晕元素的大小和透明度可以产生额外的细微效果。这个变化通过基于投射光位置与屏幕中心的距离缩放光晕元素的大小和 alpha 值来实现。光源距中心越远，元素越小越透明；光源距中心越近，元素越大越不透明。

上述方法的例子见彩图 4。

5.0.2 实现

上述方法可归结为对每个镜头光晕元素执行以下步骤：

（1）确定光晕元素的位置和大小；

（2）确定元素的纹理、颜色和透明度；

（3）采用计算出的属性，以 2D 子画面来渲染元素。

在这个实现方法中，光晕是元素的集合。每个元素都有以下静态属性：

- 纹理，可用的纹理（形状）。
- 距离，沿光源到屏幕中心直线的成比例的距离。
- 大小，（缩放前）元素的规格化大小。
- 颜色，在渲染时用于元素着色的 RGB 颜色。
- alpha，（alpha 缩放前）元素的透明度。

光晕还有个总体的缩放因子和最大尺寸，用于在渲染时控制元素的大小。这些属性都是在初始化的时候确定的。在演示代码中，属性可以随即确定，也可以从光晕描述文件载入。

在渲染过程中，每个镜头光晕的动态特性通过其静态特性与屏幕中光源的位置计算得出。光晕的纹理和颜色不会受影响，但位置、大小和 alpha 级别都将基于摄像机到光源的相对运动而变化。

镜头光晕特效的渲染伪码如下：

```
function renderflare:
    flare        // flare object to be rendered
    (lx,ly)      // projected position of light on screen
    (cx,cy)      // center of flare (normally center of screen)
{
    // Compute how far off-center the flare source is.
    maxflaredist = sqrt(cx^2 + cy^2)
    flaredist = sqrt((lx - cx)^2 + (ly - cy)^2)

    // Determine overall scaling based on off-center distance.
    distancescale = (maxflaredist - flaredist)/maxflaredist

    // Flare is rendered along a line from (lx,ly) to a
// point opposite it across the center point.
    dx = cx + (cx - lx)
    dy = cy + (cy - ly)

    for each element in flare
    {
// Position is interpolated between (lx,ly) and
// (dx,dy).
px = (1 - element.distance)*lx + element.distance*dx
        py = (1 - element.distance)*ly + element.distance*dy

        // Size of element depends on its scale, distance
// scaling, and overall scale of the flare itself.
        width = element.size * distancescale * flare.scale

        // Width gets clamped, so the off-axis flares keep a
// good size without letting the centered elements
// get too big.
        if (width > flare.maxsize)
```

```
                width = flare.maxsize

            // Flare elements are square (round) so height is
    // just width scaled by aspect ratio.
            height = width * aspectratio

            // Alpha is based on element alpha and distance scale.
            alpha = element.alpha * distancescale

            // Draw the element's texture with computed
    // properties.
            drawrectangle(    element.texture, element.colour,
    alpha, px, py, width, height )
        }
    }
```

5.0.3 源代码

镜头光晕演示示例包括 OpenGL 源代码和一个可执行的 Windows 程序。源代码分为一个 API 和使用该 API 的例子代码。API 包含定义光晕属性的结构和以下函数：

- FLARE_initialize：根据调用程序列出的属性，初始化光晕元素。
- FLARE_randomize：使用随即属性生成一系列光晕元素。
- FLARE_render：在给定屏幕位置渲染光晕。

演示示例使用上述函数，通过鼠标控制产生镜头光晕特效。鼠标光标作为光源在屏幕上的位置。镜头光晕可以随即生成，也可以从文件载入。演示示例中的 README.TXT 讲述了关于该演示界面的详细信息。