

【后处理】景深 (Depth Of Field)

笔记本： 后处理

创建时间： 2019/6/16 15:27

更新时间： 2019/6/16 16:53

作者： yuhanyou@vip.qq.com

背景介绍

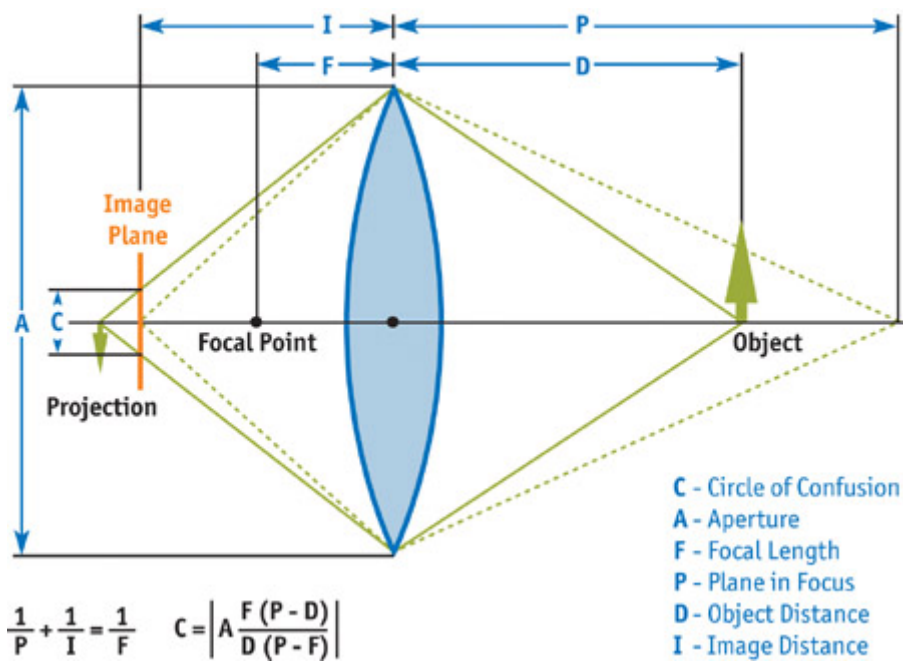


一、什么是景深？

景深 (DepthOfField, DOF)，也叫有效焦距范围，指场景中最近和最远的物体之间出现的可接受的清晰图像的现象。我们经常在生活中拿着相机对着花草或者人物进行特写时，都会出现背景虚化来凸显主体的情况。游戏中也会采用景深的情况，比如CutScene对话时，人物镜头拉近，创角捏脸等。

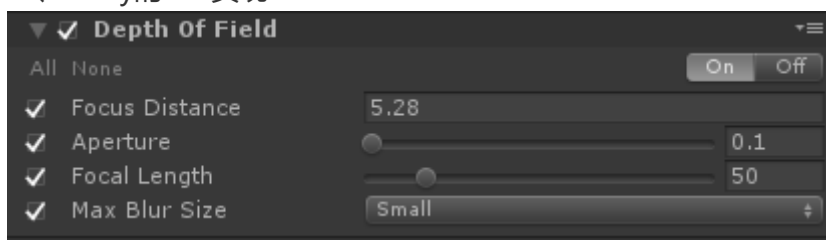
二、为什么会产生景深？

景深是由透镜的物理性质引起的，为了使光线经过相机镜头汇聚到胶片上一点，物体需要离镜头一定距离。不在这个距离的任何物体汇聚到胶片上都是一个区域，而这个区域就是**弥散圆 (Circle of confusion, CoC)**。随着镜头和物体的距离增大，CoC的直径也会随着增加。但CoC有一个范围，人眼看起来都是清晰的图像；而不在这个范围时候，我们称之为失焦。



算法实现

一、Unity的Dof实现？



- Focus Distance：焦点距离。物体到焦点的距离。
- Aperture：光圈孔径。即相机的f-number，值越小；景深越浅。
- Focal length：焦距。镜头和胶片之间的距离。数值越大，景深越浅。
- Max Blur Size：模糊半径。

DepthOfField.Shader主要可分为五个Pass：



第一步，计算弥散圆：

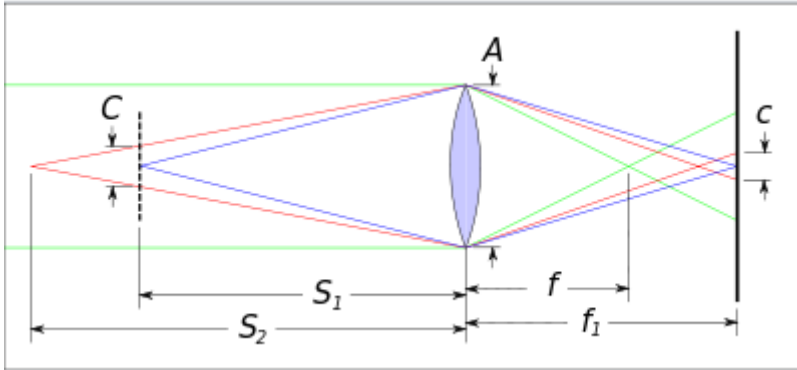
```

half4 FragCoC(VaryingsDefault i) : SV_Target
{
    float depth = LinearEyeDepth(SAMPLE_DEPTH_TEXTURE(_CameraDepthTexture,
    sampler_CameraDepthTexture, i.texcoordStereo));
    half coc = (depth - _Distance) * _LensCoeff / max(depth, 1e-5);
    return saturate(coc * 0.5 * _RcpMaxCoC + 0.5);
}
  
```

Unity计算CoC的公示是基于35mm相机公式：

$$c = \frac{fA}{S_1 - f} = \frac{f^2}{N(S_1 - f)}$$

其中公式在[Wiki中有推导过程](#)，但Unity还增加了最大弥散圆半径来Trick控制效果。



上游基于物理相机的Unity DOF效果有两个性能问题：

- 获取深度图场景再画一次，造成Draw Call消耗翻倍。
- Bokeh Filter：为了尽可能接近Disk光圈效果，滤波核半径过大，纹理采样次数过多等消耗。

二、如何优化基于手机平台的DOF效果？

1. 基于Dof Stencil的效果？

在[这篇文章](#)实现了基于特定形状的Blur效果，且《崩坏3》手游中也没有使用真正意义上的景深，那是否可以从这个角度入手呢？

如何获取标记任意区域的清晰图域？

Ans：

- 这个物体在绘制时候，需要用Buffer或者RT标记此像素区域为清晰图像？
- Unity的CommandBuffer机制，可在任意相机CameraEvent事件中插入绘制事件执行。

```
this.maskWriterCmdBuf.SetRenderTarget(this.maskWriterTex);
this.maskWriterCmdBuf.ClearRenderTarget(true, true, Color.black);

var objEnumerator = this.maskGameObjDict.GetEnumerator();
while (objEnumerator.MoveNext())
{
    foreach (Renderer r in
objEnumerator.Current.Value.GetComponentsInChildren<Renderer>())
    {
        this.maskWriterCmdBuf.DrawRenderer(r, this.maskWriterMat);
    }
}
objEnumerator.Dispose();
```

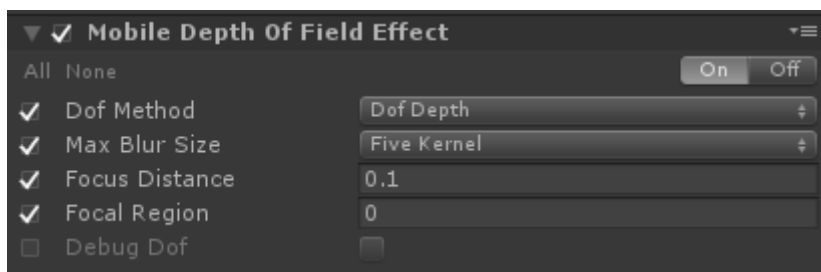
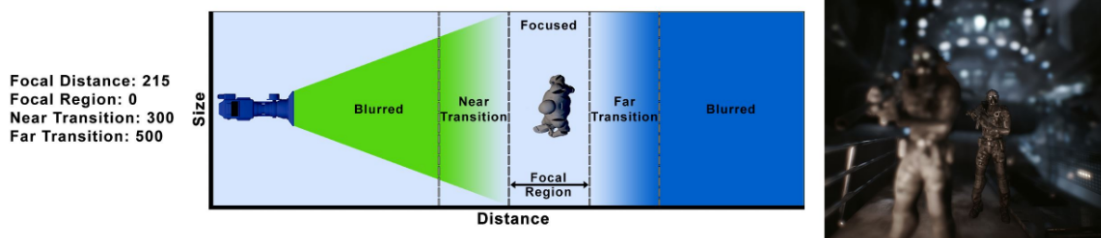
```

this.maskWriterCmdBuf.SetGlobalTexture("_MaskBlurTex", this.maskWriterTex);
if (this.mainCamera != null)
{
    this.mainCamera.AddCommandBuffer(CameraEvent.BeforeForwardOpaque,
    this.maskWriterCmdBuf);
}

```

2. 基于Dof Depth的效果？

除了上面的方案，美术还可能需要基于距离的方案？



第一步：提供尽量少参数来表达此景深示意图。

- Focus Distance：对焦距离。
- Focal Region：对焦区域。区域中，清晰图像；区域外，模糊图像。

第二步：通过当前场景深度值与区域进行比较。

```

half4 FragDofCoc(VertexOutput_Img input) : SV_TARGET
{
    float depth = LinearEyeDepth( SAMPLE_DEPTH_TEXTURE(_CameraDepthTexture,
    sampler_CameraDepthTexture, input.texcoord).r );
    half Coc = abs(depth - uDofFocusDistance) < uDofFocalRegion ? 1 : 0;
    return Coc;
}

```

第三步：优化Depth 获取。

- 避免场景再画一次得到深度图，修改渲染管线，是场景深度值保存在ColorBuffer中的A通道。
- 拷贝Pass：执行CommandBuffer事件，拷贝默认depthbuffer到RT中。

未来展望

基于物理的手机平台的景深效果。

参考资料

- 【1】 : [Wiki: Depth of Field](#)
- 【2】 : [Wiki: Circle of Confusion](#)
- 【3】 : [Depth of Field: A Survey of Techniques](#)
- 【4】 : [基于物理的景深效果](#)
- 【5】 : [Image Effects扩展——支持Clip的Blur](#)
- 【6】 : [米哈游技术总监首次分享：移动端高品质卡通渲染的实现与优化方案](#)
- 【7】 : [UE4:移动平台景深效果](#)
- 【8】 : [UnityTutorial : Depth of Field](#)
- 【9】 : [The Skylanders SWAP Force Depth-of-Field Shader](#)