



唐老狮系列教程

边缘检测基本原理



唐老狮系列教程-边缘检测基本原理

| 主要讲解内容



唐老狮系列教程-边缘检测基本原理

主要讲解内容

1. 边缘检测效果是什么
2. 边缘检测效果的基本原理
3. 如何得到当前像素周围8个像素位置



唐老狮系列教程-边缘检测基本原理

| 边缘检测效果是什么



唐老狮系列教程-边缘检测基本原理

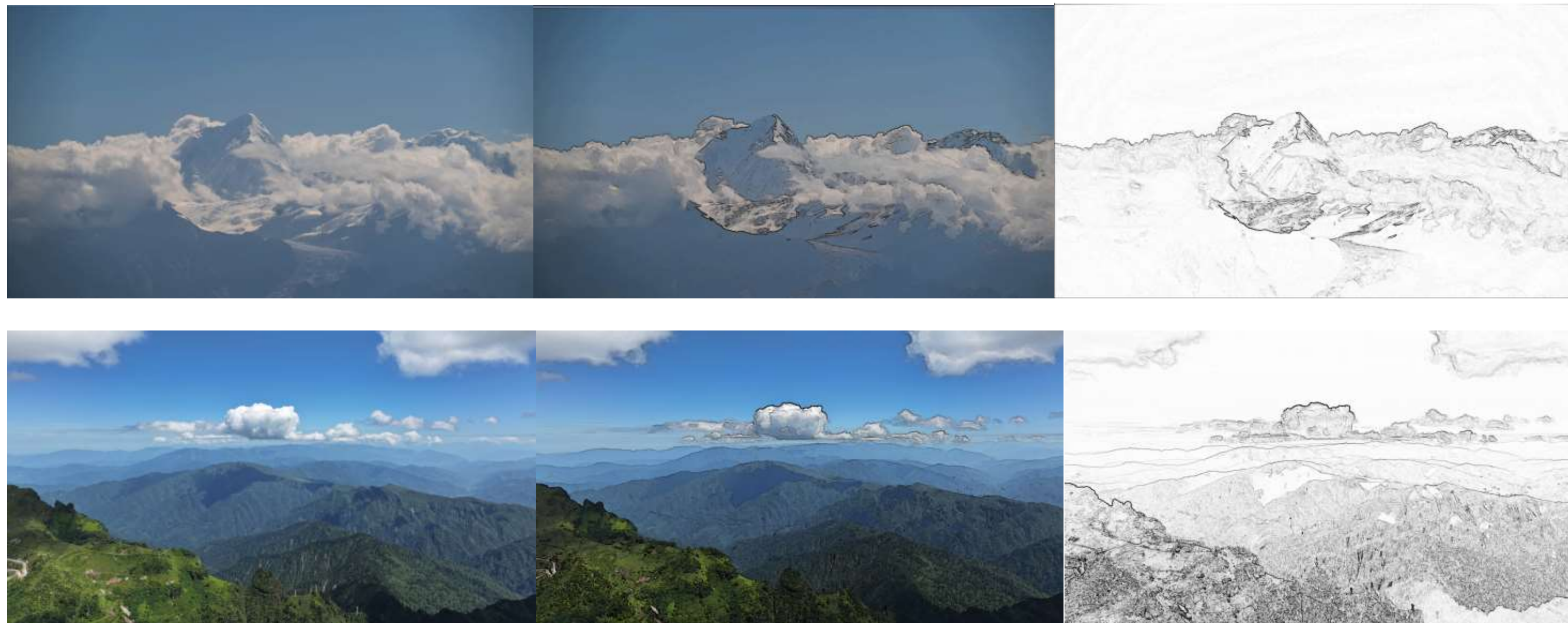
边缘检测效果是什么

边缘检测效果，是一种

用于**突出图像中的边缘，使物体的轮廓更加明显**的图像处理技术

边缘检测的主要目的是找到图像中亮度变化显著的区域，这些区域通常对应于物体的边界

边缘检测相当于利用 Shader 代码自动给屏幕图像进行描边处理





唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理



唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理

一句话概括Unity Shader中实现边缘检测效果的基本原理：

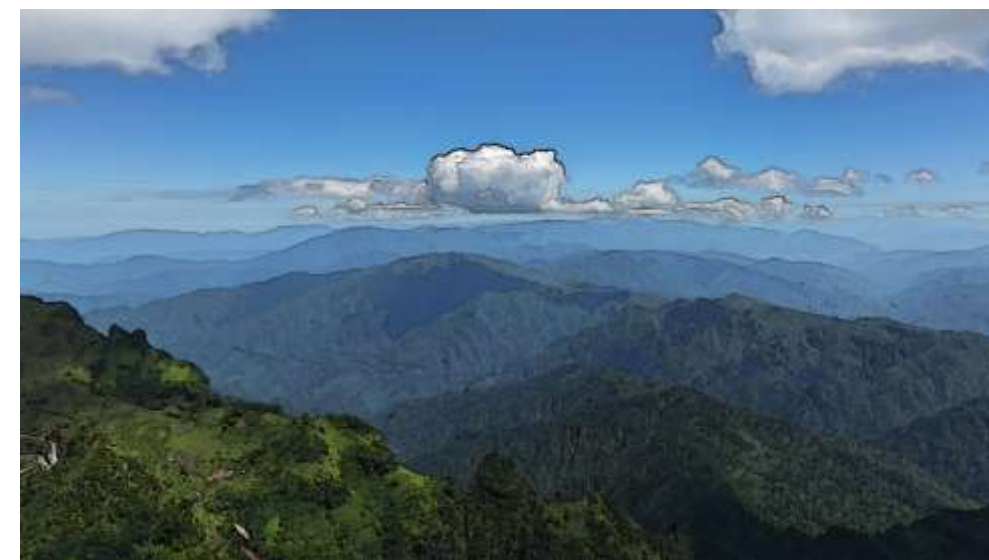
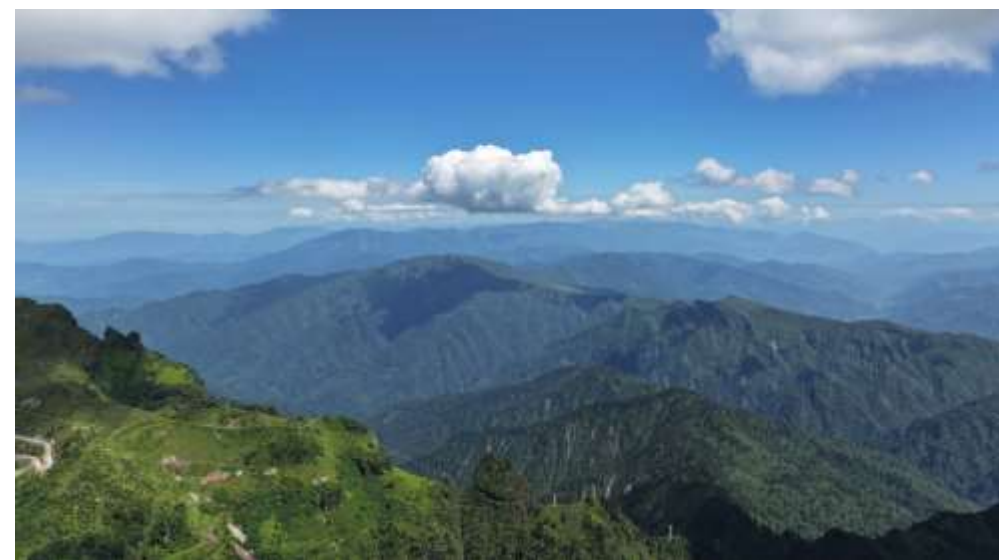
计算每个像素的灰度值，用灰度值结合卷积核进行卷积运算，得到该像素的梯度值

梯度值越大越靠近边界，越趋近于描边颜色

梯度值越小表明不是边界位置，越趋近于原始颜色

关键知识点：

灰度值、卷积、卷积核、梯度值





唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理 —— 灰度值

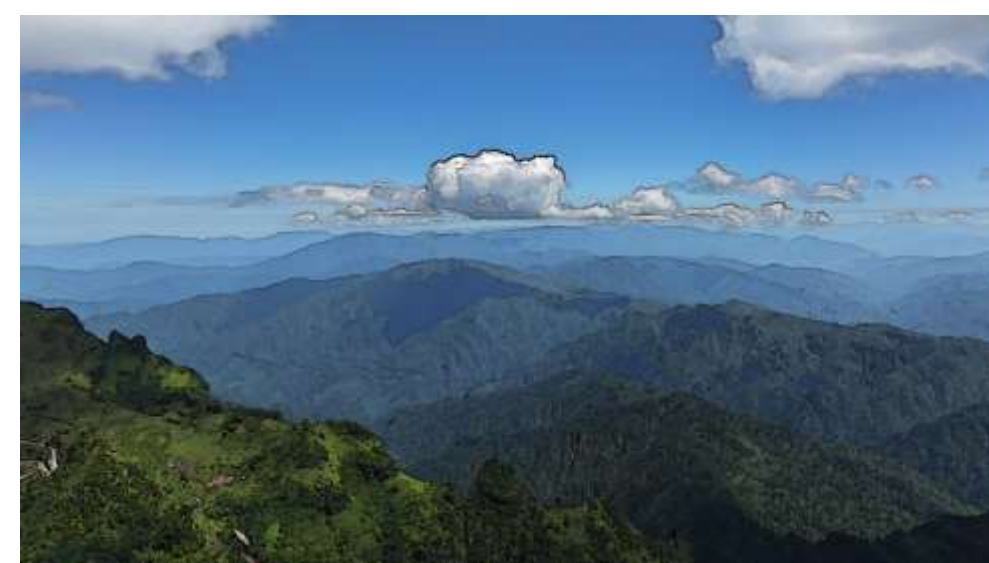
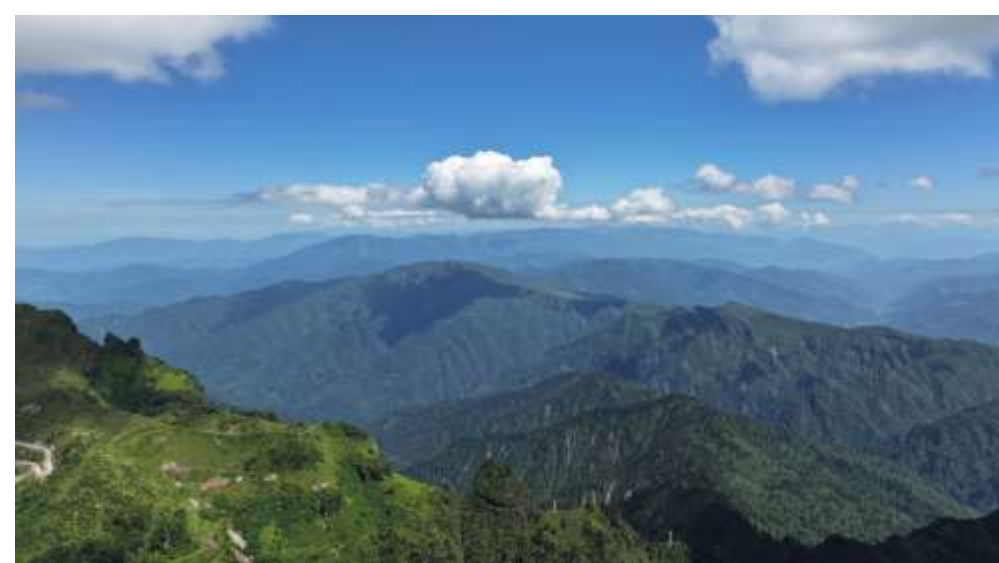
由于人眼对不同颜色的敏感度不同，所以在计算平均值时不会直接使用算数平均 $(R+G+B)/3$

在图形学中我们一般使用加权平均法来计算灰度值

所谓加权平均法就是通过对不同数据分配不同权重，计算出更符合实际情况的平均值

下面是基于 **Rec. 709标准** 计算的灰度值（高清电视和许多数字图像格式中常用的标准）

$$\text{灰度值 } L = 0.2126 * R + 0.7152 * G + 0.0722 * B$$





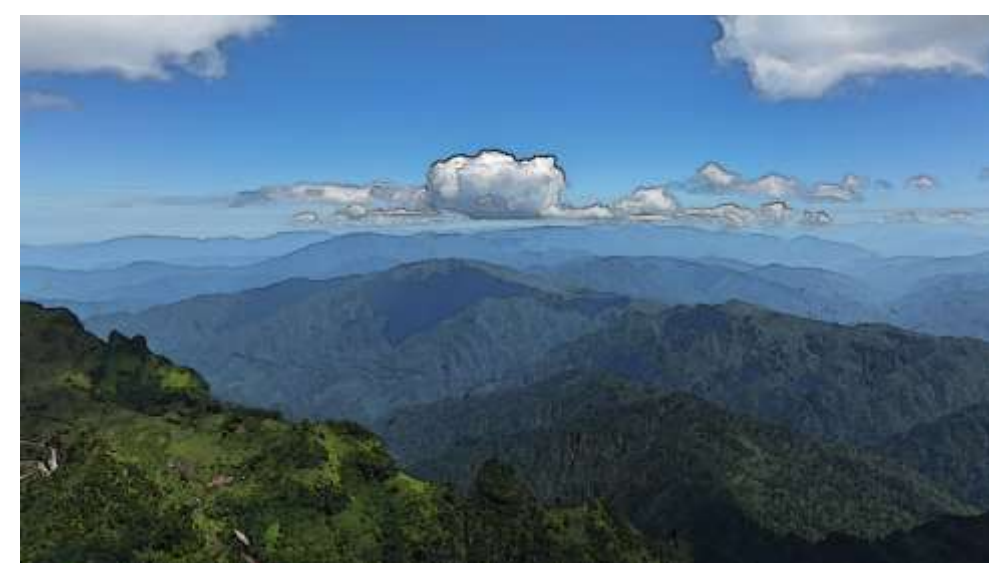
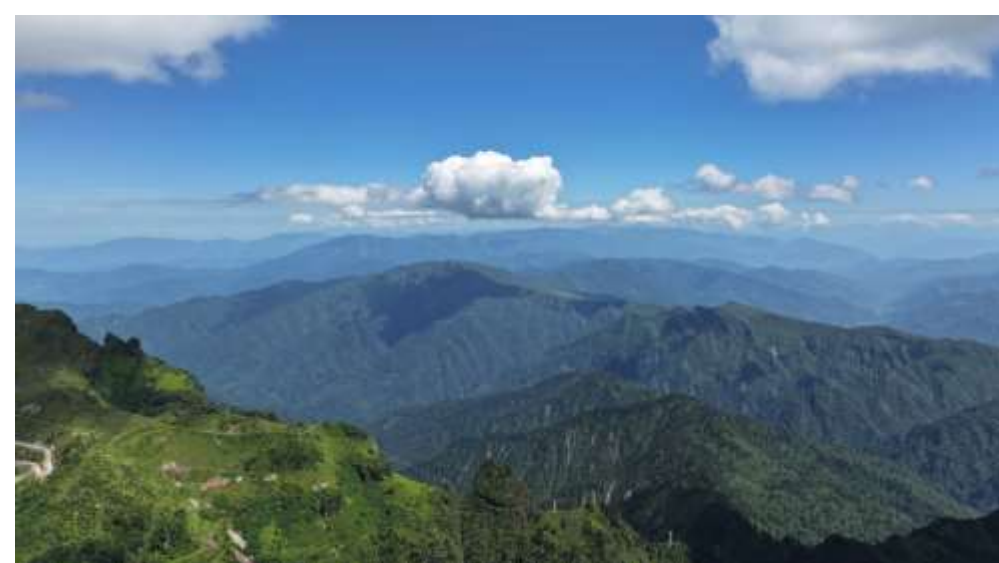
唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理 —— 卷积、卷积核、梯度值

卷积是一种数学计算方式

我们首先通过一个比喻来理解卷积在边缘检测中的作用

它就像是要用一个**放大镜（卷积核）**在图片上移动，**放大镜（卷积核）**的作用是帮助我们看到图片上的细微变化。当我们用这个**放大镜（卷积核）**扫描整张图片时，它能帮助我们**发现**图片上**哪些地方颜色变化突然，这些突然变化的地方往往就是物体的边缘了**





唐老师系列教程-边缘检测基本原理

边缘检测效果的基本原理 —— 卷积、卷积核、梯度值

假设我们有一张 5x5 的图像，每一个格子代表一个像素

格子中的数据表示该像素的灰度值

假设我们有一个 3x3 的卷积核（放大镜）

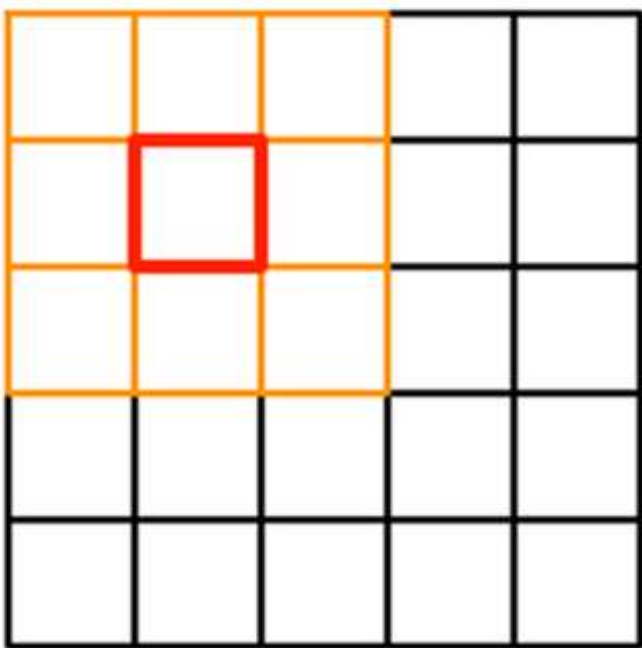
1	2	3	4	5
6	7	8	9	8
7	8	6	5	4
1	0	1	2	3
2	3	4	5	6

一个5x5的图像

如果你想要求出图中红色格子的梯度值（值越大表示越靠近边缘）

1	0	-1
1	0	-1
1	0	-1

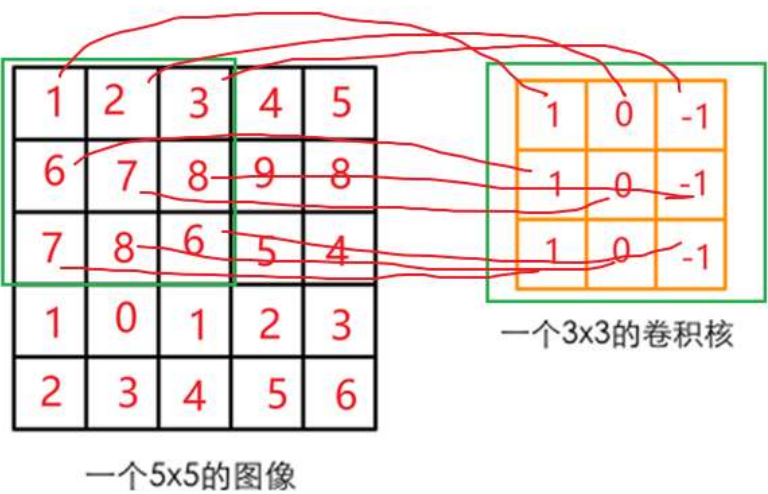
一个3x3的卷积核



进行卷积计算

那么只需要用 卷积核 和 对应位置像素的灰度值

进行如下计算：



$$\begin{aligned} &1*1 + 2*0 + 3*-1 + \\ &6*1 + 7*0 + 8*-1 + \\ &7*1 + 8*0 + 6*-1 = -3 \end{aligned}$$

最终算出来的结果就表示该像素的梯度值，我们便可以用该值决定边缘效果了



唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理 —— 卷积、卷积核、梯度值

从卷积的计算方式我们可以得知，其中**卷积核（也被称为边缘检测因子）**是非常重要的一个元素，在图形学中，有**三种常用的卷积核（边缘检测因子）**他们分别是：

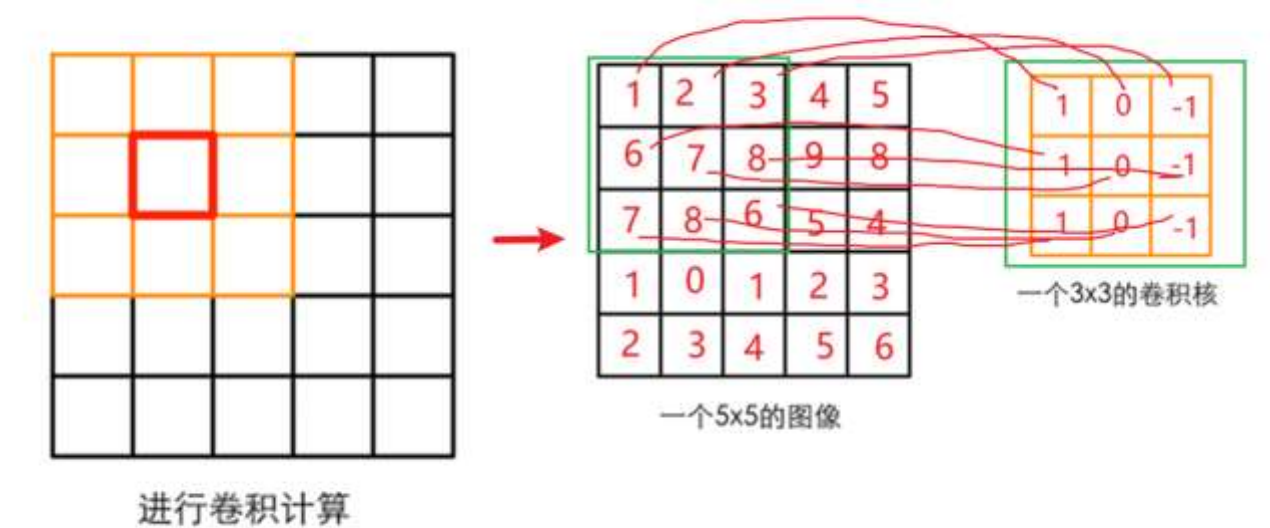
Roberts 算子：由拉里·罗伯茨(Larry Roberts)于1965年提出

Prewitt 算子：由约翰·普雷维特(John Prewitt)于1970年提出

Sobel 算子：由欧文·索贝尔(Irwin Sobel)于1968年提出

他们各有千秋，但是在图形学中**最常用的还是Sobel算子**

因为它更适合高精度的边缘检测



Roberts

-1	0	0	-1
0	1	1	0
Gx		Gy	

Prewitt

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1
Gx			Gy		

Sobel

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1
Gx			Gy		



唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理 —— 卷积、卷积核、梯度值

我们可以看到三种算子都包含了**两个方向的卷积核**

他们**分别用来检测水平和竖直方向上的边缘信息**

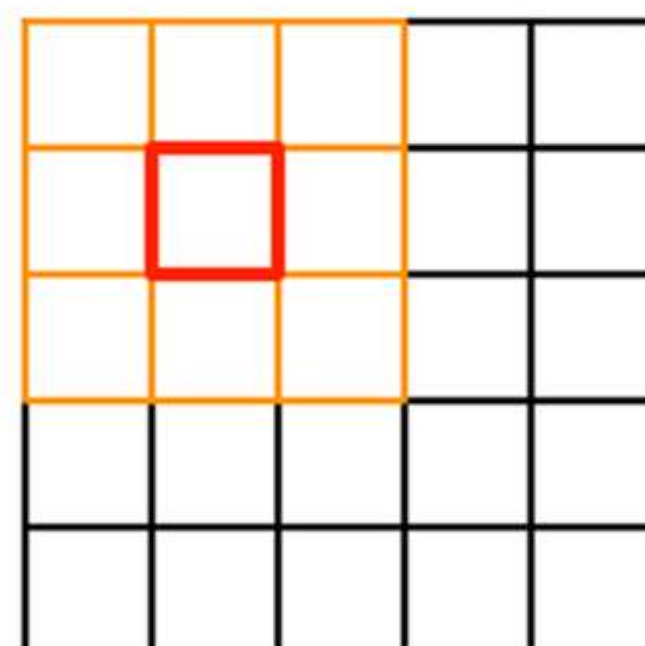
在**边缘检测的卷积计算时**，只需要**对每个像素进行两次卷积计算**即可

这样就可以**得到两个方向的梯度值 G_x 和 G_y**

而**该像素**的整体**梯度值 $G = \text{abs}(G_x) + \text{abs}(G_y)$**

1	2	3	4	5
6	7	8	9	8
7	8	6	5	4
1	0	1	2	3
2	3	4	5	6

一个5x5的图像



进行卷积计算

Sobel

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1
G_x			G_y		



唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理 —— 卷积、卷积核、梯度值

现在我们已经知道了通过卷积获取梯度值的基本原理

那么我们只需要定义一个描边颜色，利用Shader中的内置函数lerp

在原始颜色和描边颜色之间利用梯度值进行插值即可

最终颜色 = lerp（原始颜色，描边颜色，梯度值）

梯度值越大表明越接近边缘，则颜色越接近描边颜色；反之越接近原始颜色





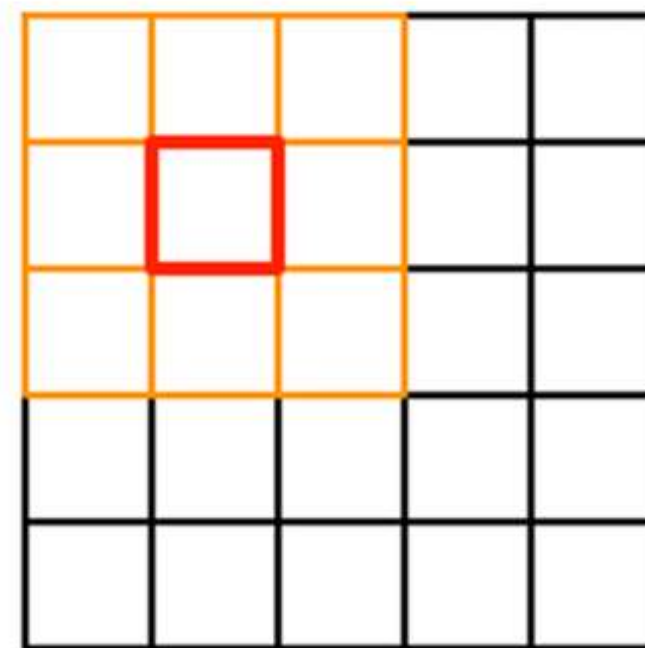
唐老狮系列教程-边缘检测基本原理

边缘检测效果的基本原理

1. 得到 当前像素以及其 上下左右、左上左下、右上右下共9个像素的灰度值
2. 用这9个灰度值和 Sobel算子 进行卷积计算得到梯度值 $G = \text{abs}(Gx) + \text{abs}(Gy)$
3. 最终颜色 = lerp (原始颜色, 描边颜色, 梯度值)

1	2	3	4	5
6	7	8	9	8
7	8	6	5	4
1	0	1	2	3
2	3	4	5	6

一个5x5的图像



进行卷积计算

Sobel

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Gx

Gy



唐老狮系列教程-边缘检测基本原理

如何得到当前像素周围8个像素位置



唐老狮系列教程-边缘检测基本原理

如何得到当前像素周围8个像素位置

想要获取当前像素周围8个像素的位置，我们需要补充一个知识点
即：

Unity 提供给我们用于访问纹理对应的每个纹素（像素）的大小 的变量
float4 纹理名_TexelSize

类似 纹理名_ST（用于获取纹理缩放偏移的变量）

其中的xyzw分别代表（假设纹理宽高为 **1024 * 768**）

x: 1/纹理宽度 = 1/1024

y: 1/纹理高度 = 1/768

z: 纹理宽度 = 1024

w: 纹理高度 = 768

```
sampler2D _MainTex;  
float4 _MainTex_ST;  
half4 _MainTex_TexelSize;
```

1	2	3	4	5
6	7	8	9	8
7	8	6	5	4
1	0	1	2	3
2	3	4	5	6

一个5x5的图像



唐老狮系列教程-边缘检测基本原理

如何得到当前像素周围8个像素位置

我们可以利用 float4 纹理名_TexelSize 纹素 信息得到当前像素周围8个像素位置
可以**进行uv坐标偏移计算**，在 顶点着色器函数 或者 片元着色器函数 中计算都行
但是**建议在顶点着色器函数中计算，可以节约计算量**

片元着色器中直接使用插值的结果也不会影响纹理坐标的计算结果

```
half2 uv = v.texcoord;

o.uv[0] = uv + _MainTex_TexelSize.xy * half2(-1, -1);
o.uv[1] = uv + _MainTex_TexelSize.xy * half2(0, -1);
o.uv[2] = uv + _MainTex_TexelSize.xy * half2(1, -1);
o.uv[3] = uv + _MainTex_TexelSize.xy * half2(-1, 0);
o.uv[4] = uv + _MainTex_TexelSize.xy * half2(0, 0);
o.uv[5] = uv + _MainTex_TexelSize.xy * half2(1, 0);
o.uv[6] = uv + _MainTex_TexelSize.xy * half2(-1, 1);
o.uv[7] = uv + _MainTex_TexelSize.xy * half2(0, 1);
o.uv[8] = uv + _MainTex_TexelSize.xy * half2(1, 1);
```

1	2	3	4	5
6	7	8	9	8
7	8	6	5	4
1	0	1	2	3
2	3	4	5	6

一个5x5的图像



唐老狮系列教程-边缘检测基本原理

| 总结



唐老狮系列教程-边缘检测基本原理

主要讲解内容

1. 边缘检测效果是什么

是一种用于突出图像中的边缘，使物体的轮廓更加明显的图像处理技术

利用 Shader 代码自动给屏幕图像进行描边处理

2. 边缘检测效果的基本原理

2-1 得到 当前像素以及其 上下左右、左上左下、右上右下共9个像素的灰度值

2-2 用这9个灰度值和 Sobel算子 进行卷积计算得到梯度值 $G = \text{abs}(Gx) + \text{abs}(Gy)$

2-3 最终颜色 = lerp (原始颜色, 描边颜色, 梯度值)

3. 如何得到当前像素周围8个像素位置

利用 float4 纹理名_TexelSize 纹素 信息得到当前像素周围8个像素位置



唐老狮系列教程

Thank

谢谢您的聆听