

HW6 作业报告

邱荻 2000012852 信息科学技术学院

#基础型

我编程实现了一种照片尺寸自适应的暗角(Vignette)模拟生成算法，并在多个不同大小的照片上作测试验证，效果很好。

方法:

1. 计算图像中心到四个角的距离

得到图像的高与宽，然后计算图像中心到四个角的距离，后续的亮度调整的理念是离中心距离越远的点调整得越暗。

2. 生成距离网格

接下来，我们使用 `meshgrid` 函数生成一个与输入图像大小相同的网格，其中每个点表示该点到图像中心的距离。

3. 计算亮度调整系数

根据距离网格，计算每个像素点的亮度调整系数。这些系数用于控制每个像素点的亮度，距离中心点越远的像素点亮度越低。具体地，系数是这样

`vignette = 1 - (distances / max_distance);`

4. 应用亮度调整系数

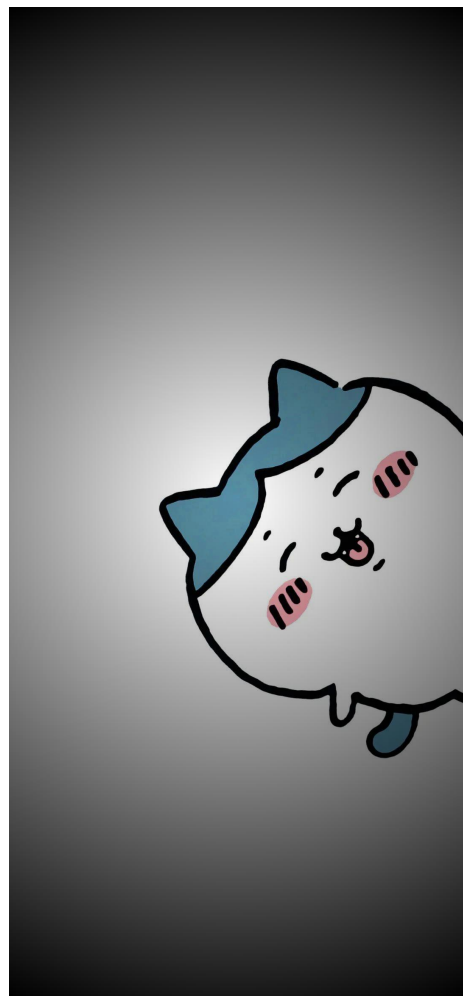
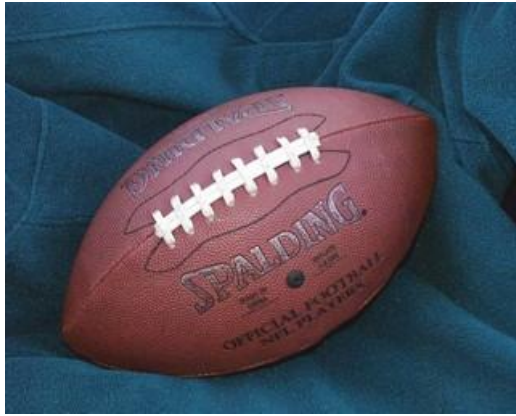
最后，我们将计算得到的亮度调整系数应用到输入图像的每个通道上，使用 MATLAB 的 `bsxfun` 函数实现通道之间的逐元素相乘，就得到了最终的暗角效果图像。

代码如下:

```
main.m × +
/MATLAB Drive/hw6/main.m
1 function output_image = generate_vignette(input_image)
2 %得到图像高和宽 以及中心的位置
3 [height, width, ~] = size(input_image);
4 center = [width/2, height/2];
5
6 %计算图像中心到四个角的距离
7 max_distance = norm(center);
8
9 %生成一个网格 每个点表示该点到图像中心的距离
10 [x, y] = meshgrid(1:width, 1:height);
11 distances = sqrt((x - center(1)).^2 + (y - center(2)).^2);
12
13 %根据距离计算每个点的亮度调整系数 距离越远越黑
14 vignette = 1 - (distances / max_distance);
15 vignette = max(0, min(vignette, 1));
16
17 %将uint8类型的图像转换为double类型 归一化到[0,1] (不然报错)
18 input_image = double(input_image) / 255.0;
19
20
21 %调整图像 离中心越远越黑
22 output_image = bsxfun(@times, input_image, vignette);
23 end
24
25 input_image = imread('cat.png');
26
27 output_image = generate_vignette(input_image);
28
29 imwrite(output_image, 'cat_vignette.png');
30
```

结果:





可以看到，结果非常好!