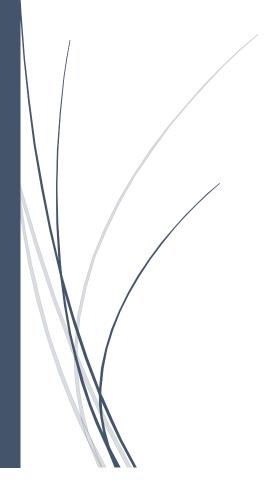
HOMEWORK6



Cecil Wang MICROSOFT

目录

一、作业描述	2
二、实现细节	. 2
2.1 题目分析	
2.1 代码细节	. 3
三、字吟结果	

一、作业描述

编程实现一种色调转移或饱和度调整算法。

二、实现细节

2.1 题目分析

要完成色调转移与饱和度调整,首先需要将图片的色调和饱和度计算出来,由此可以想到利用 HSV 色彩空间模型。所谓 HSV 就是 Hue(色调)、Saturation(饱和度)、Value(明度),在空间上标识 HSV 是一种六角锥体的模型。一般情况下,Hue 的取值范围为[0,360],而 Saturation 和 Value 的取值范围是[0,1]。

具体的 RBG 与 HSV 相互转化的公式如下:

RGB->HSV

$$V = \max(R, G, B)$$

$$S = \begin{cases} 0 & , V = 0 \\ \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} & , \text{ otherwise} \end{cases}$$

$$H = \begin{cases} 0 + \frac{60*(G-B)}{\max(R,G,B) - \min(R,G,B)} & , R = \max(R,G,B) \\ 120 + \frac{60*(B-R)}{\max(R,G,B) - \min(R,G,B)} & , G = \max(R,G,B) \\ 240 + \frac{60*(R-G)}{\max(R,G,B) - \min(R,G,B)} & , B = \max(R,G,B) \end{cases}$$

注:将H的值归一化到[0,360]

HSV ->RGB

$$flag = \left(\frac{H}{60}\right) \mod 6$$

$$f = \frac{H}{60} - \left(\frac{H}{60}\right) \mod 6$$

$$p = v - (1 - s)$$

$$q = v - (1 - f * s)$$

$$t = v - (1 - (1 - f) * s)$$

$$(R, G, B) = \begin{cases} (v, t, p) & flag = 0 \\ (q, v, p) & flag = 1 \\ (p, v, t) & flag = 2 \\ (p, q, v) & flag = 3 \\ (r, p, v) & flag = 4 \\ (v, p, q) & flag = 5 \end{cases}$$

2.1 代码细节

对于 HSV 转 RBG 最后的赋值阶段由于有 6 种情况,编码繁琐,所以我将 p,q,t,v 存入一个列向量 val 中,每次通过 index 数组来决定具体的赋值顺序,具体如下:

```
index=[
   4 3 1
    2 4 1
    3 1 4
    4 1 2
for i = 1:height
    for j = 1:width
        h = floor(HSV(i,j,1) / 60);
        f = HSV(i,j,1) / 60 - h;
       val(1) = HSV(i,j,3) * (1 - HSV(i,j,2));
       val(2) = HSV(i,j,3) * (1 - f * HSV(i,j,2));
       val(3) = HSV(i,j,3) * (1 - (1 - f) * HSV(i,j,2));
       val(4) = HSV(i,j,3);
        RGB(i,j,:) = val(index(h+1,:));
    End
End
```

对于色调转移,就是整体将 Hue 每个像素进行 加 delta 然后 mod 360,即

 $Hue = (Hue + delta) \mod 360$

而对于饱和度,delta 代表了百分比,然后夹逼到[0,1],即

Saturation =

$$Saturation = \begin{cases} 1 & \text{,} Saturation * (1 + delta) > 1 \\ Saturation * (1 + delta) & \text{,} otherwise \\ 0 & \text{,} Saturation * (1 + delta) < 0 \end{cases}$$

三、实验结果

