

作业11：图像自动白平衡

作业目的：

了解图像色彩处理的典型应用

作业内容：

对图像进行自动白平衡处理，消除光照的影响，还原图像中对象本来的色彩

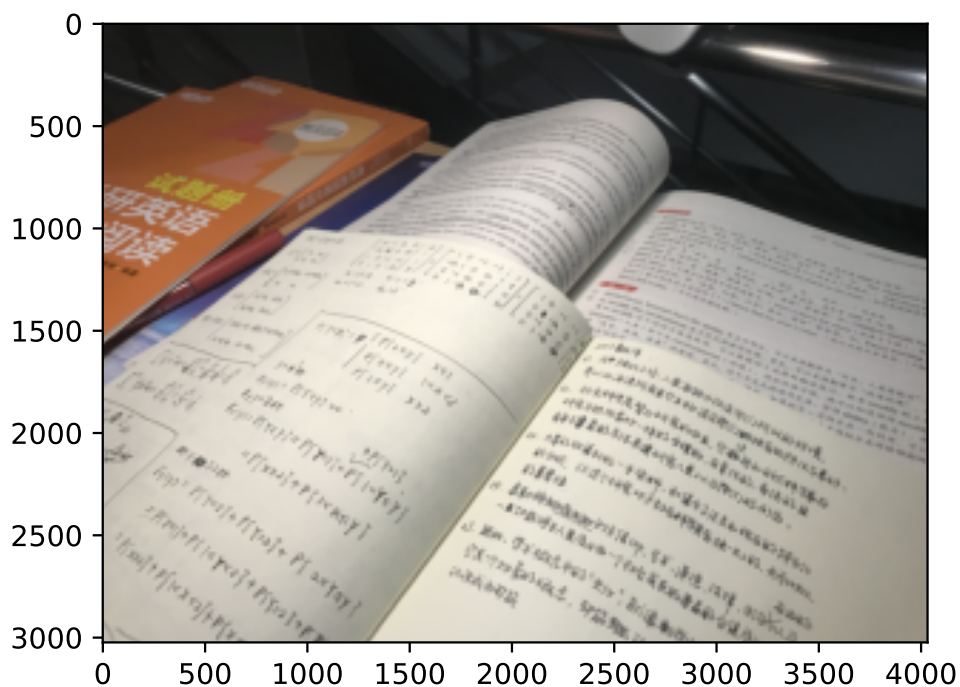
```
In [9]: from skimage import io, img_as_float
import matplotlib.pyplot as plt
import numpy as np
```

作业要求：

(1) 拍摄或选择一张受环境光照影响的图片

```
In [10]: plt.imshow(io.imread('./img/example.png'))
```

```
Out[10]: <matplotlib.image.AxesImage at 0x7fe16a0bd310>
```



(2) 阅读附件中的中文文献，选择一种自动白平衡算法用python进行实现，观察处理效果

```
In [11]: # 灰度世界算法 GW ( Gray world Assumption)
def balance(img):
    r = img[:, :, 0]
    g = img[:, :, 1]
    b = img[:, :, 2]
    mr = np.mean(r)
    mg = np.mean(g)
    mb = np.mean(b)
    k = (mr + mg + mb) / 3
    kb = k / mb
    kg = k / mg
    kr = k / mr

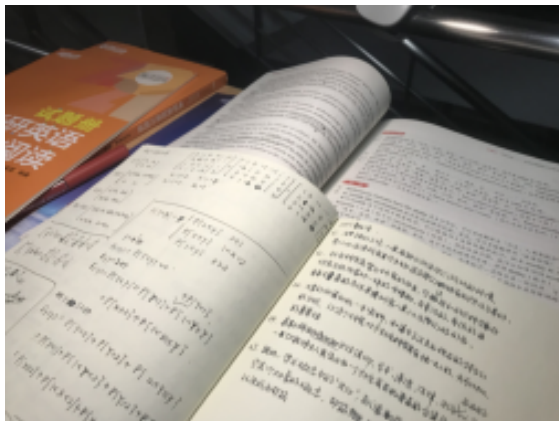
    R = r * kr
    G = g * kg
    B = b * kb
    balanced = np.dstack((R, G, B))
    return balanced
```

```
In [12]: img = img_as_float(io.imread('./img/example.png'))
bal = balance(img)

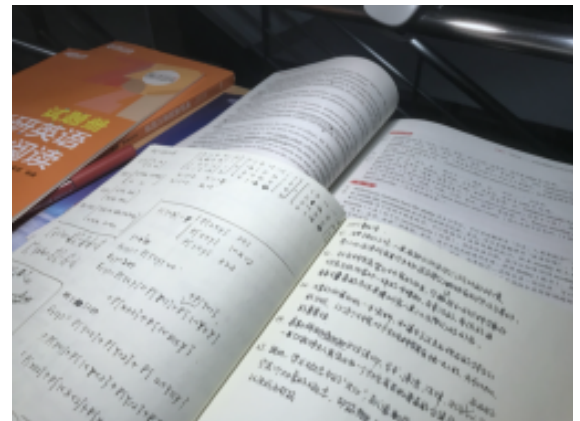
plt.figure(figsize=(10,5)) #设置窗口大小
plt.subplot(1,2,1), plt.title('original image')
plt.imshow(img), plt.axis('off')
plt.subplot(1,2,2), plt.title('balanced image')
plt.imshow(bal), plt.axis('off')
plt.show()
```

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

original image



balanced image



In []: 从图中可以明显的看得出来，图二的色温更冷，更接近真实白光的色温，原图像我是由暖色温的灯光拍

(3) 可选作业：对同一张图片，用英文文献给出的白平衡算法进行处理，观察处理效果

作业提交：

- (1) PDF文件格式
- (2) 给出一种自动白平衡算法的原理、Python代码和效果
- (3) 可选作业：给出英文文献处理效果