

Chapter 6 Color Image Processing

李想 P12214061

2025 年 4 月 30 日

1 问题一

彩色图像进行直方图均衡化，首先想到将彩色图像的各个通道分别均衡化，但这种方法会产生错误的彩色。更合适的做法是，均匀地分布颜色亮度，而保持颜色本身（色调）不变。HSI 彩色空间是适合直方图均衡化的理想空间。

1.1 RGB to HSI

已知一幅 RGB 彩色格式的图像，每个 RGB 像素的 I 分量可由公式 (1) 得到:

$$I = \frac{1}{3}(R + G + B) \quad (1)$$

饱和度分量 S 为:

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)] \quad (2)$$

色调分量 H 为:

$$H = \begin{cases} \theta & B \leq G \\ 2\pi - \theta & B > G \end{cases} \quad (3)$$

1.2 亮度直方图均衡

将 I 分量归一化到 [0-255] 区间内，并用第三章的直方图均衡化对 I 分量单独处理，均衡前后的直方图分别如图1a和图1b 所示。均衡前的直方图集中在中等灰度区和高灰度区，CDF 曲线在中等灰度区有一个陡峭的上升趋势。均衡后的直方图则均匀分布在 [0-255] 区间内，CDF 曲线在整个区间内，近似为一条从左下角到右上角的直线。

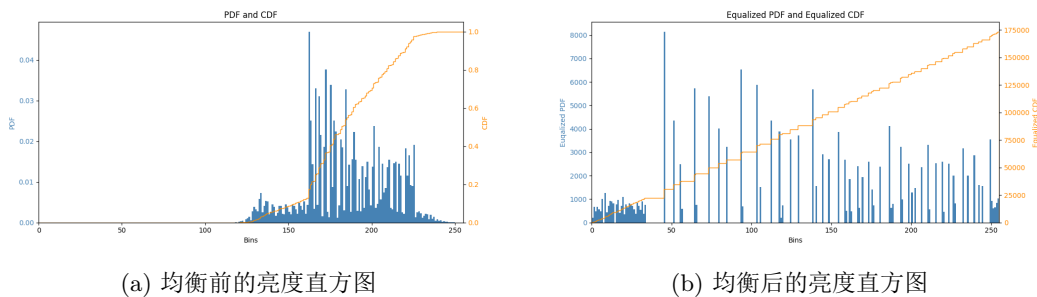


图 1: 直方图均衡前后的亮度直方图和 CDF

直方图均衡前后的 I 分量分别如图2a 和图2b所示对比观察图2a和图2b，发现均衡后的 I 分量对比度增大，阴影和高光相较于均衡前更为明显。

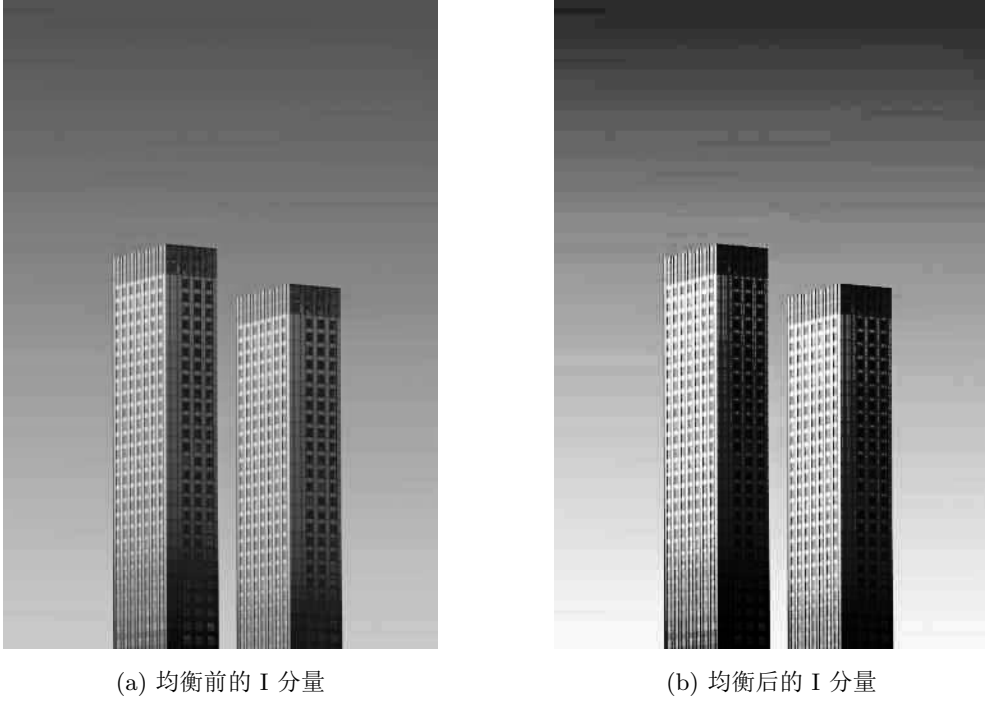


图 2: 直方图均衡前后的 I 分量

1.3 HSI to RGB

为了显示直方图均衡后的彩色图像，我们还需要将 HSI 空间的图像转换到 RGB 空间。从 HSI 空间转换到 RGB 空间的公式为式 (4)(5) (6):

当 $0 \leq H < \frac{2\pi}{3}$ 时:

$$\begin{aligned}
 B &= I(1 - S), \\
 R &= I \left[1 + \frac{S \cos H}{\cos(\frac{\pi}{3} - H)} \right], \\
 G &= 3I - (R + B).
 \end{aligned} \tag{4}$$

当 H 落在 $[\frac{2\pi}{3}, \frac{4\pi}{3})$ 时, 令 $H' = H - \frac{2\pi}{3}$, 计算公式为:

$$\begin{aligned}
 R &= I(1 - S), \\
 G &= I \left[1 + \frac{S \cos H'}{\cos(\frac{\pi}{3} - H')} \right], \\
 B &= 3I - (R + G).
 \end{aligned} \tag{5}$$

当 H 落在 $[\frac{4\pi}{3}, 2\pi)$ 时, 令 $H'' = H - \frac{4\pi}{3}$, 计算公式为:

$$\begin{aligned}
 G &= I(1 - S), \\
 B &= I \left[1 + \frac{S \cos H''}{\cos(\frac{\pi}{3} - H'')} \right], \\
 R &= 3I - (G + B).
 \end{aligned} \tag{6}$$

直方图均衡后 HSI 图像转换到 RGB 空间后, 如图3b 所示, 原始图像为图3a。对比观察图3a 和图3b, 我们发现原始图像的颜色较淡, 图像整体偏亮, 而直方图均衡后的图像, 颜色更深, 整体亮度适中。尽管亮度直方图均衡并不会改变色调值和饱和度值, 但确实会影响图像的整体颜色。改进方法是, 首先增大图像的饱和度分量, 然后再进行亮度直方图均衡化。

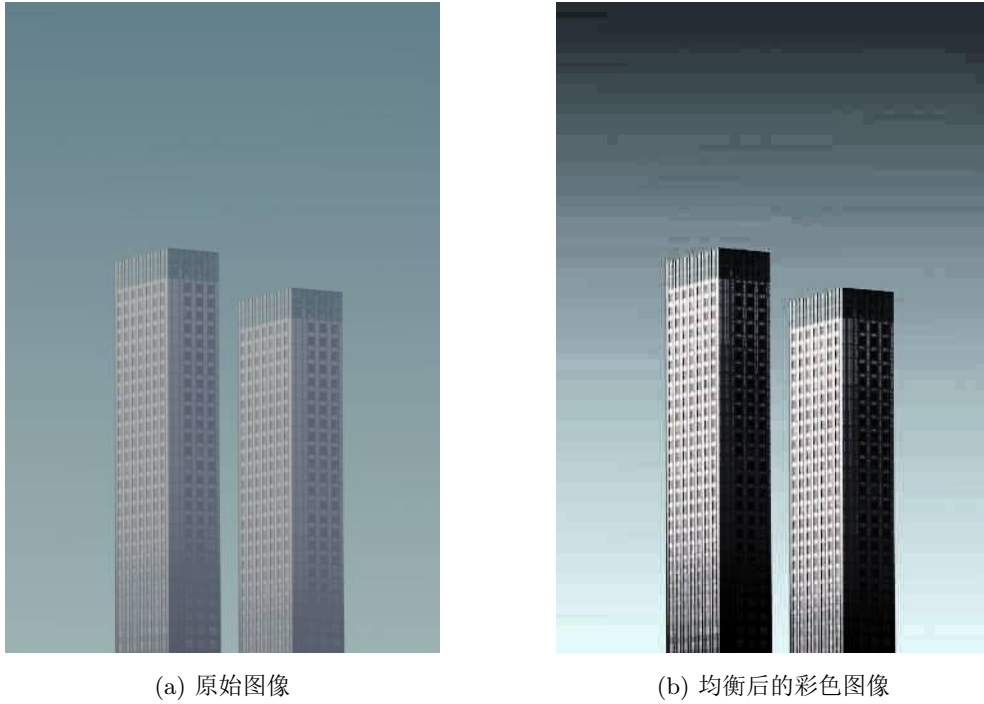


图 3: 直方图均衡前后的彩色图像

2 问题二

绿幕抠图系统由两个部分组成，前景图像和背景图像，前景图像中必须有一个物体放置在绿色的背景之上，背景图像则可以任意选择。前景图像和背景图像大小必须相同，抠像系统的输出也与前景和背景大小相同。在抠图的过程中，我们创建了一个新的图像（输出）。输出图像的像素依据下面的过程获得：

1. 对前景图像中的每个像素，提取其绿色通道的灰度值；
2. 若该灰度值大于设定阈值（如 220），则将输出图像中对应位置的像素替换为背景图像中相同位置的像素；
3. 否则，保留前景图像中对应位置的像素至输出图像。

算法流程如算法 1 所示。

Algorithm 1: 绿幕抠像算法

Input: 前景图像 F ，背景图像 B ，阈值 T

Output: 合成图像 O

```

1 foreach 像素位置  $(x, y)$  do
2    $g \leftarrow F(x, y).G$  ;                                // 获取绿色通道
3   if  $g > T$  then
4      $O(x, y) \leftarrow B(x, y)$  ;                        // 替换为背景像素
5   else
6      $O(x, y) \leftarrow F(x, y)$  ;                        // 保留前景像素

```

前景，背景和输出图像分别如图4a，图4b，图4c所示：观察输出图像，发现抠出的图像存在绿



(a) 前景



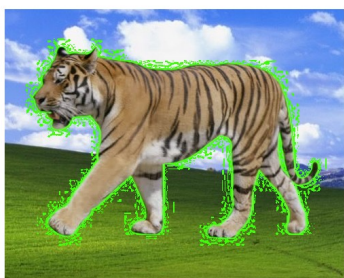
(b) 背景



(c) 输出图像 ($T=220$)

图 4: 抠像系统的前景、背景和输出

色边框，增大绿色通道的阈值，边框变厚，如图5a，减小绿色通道的阈值，绿色边框变细，但抠出的物体会存在像素缺失，如图5b，不能完美的抠出前景中的物体是该抠像算法的局限性。



(a) 阈值 $T=255$



(b) 阈值 $T=200$

图 5: 更改阈值后的输出