

图像灰度化方法

PB20030896 江鹏飞

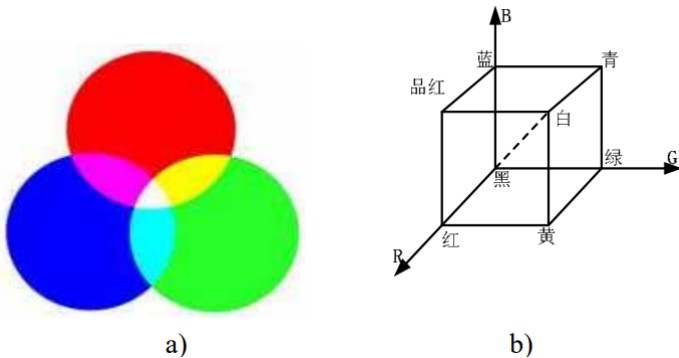
摘要：彩色图片灰度化是数字图像处理、黑白风格摄影等许多应用中常见的方法。由于在转换过程当中不可避免地会出现特征信息丢失等问题，因此，为了更好地保持输入图像的特征信息并改善灰度图像的视觉效果，学者们提出了许多彩色图像灰度化算法。本文尝试实现了几个经典的灰度化算法。

1.前言

彩色图像灰度化实质上是一个将彩色图像的三维像素矩阵经过降维生成一个二维灰度矩阵的过程。该过程中应最大程度地保留彩色图像的显著特征信息，使转换前后的图像在亮度、对比度及结构信息方面均达到最优匹配。在介绍灰度化算法之前，我们有必要了解两种色彩模型。

- RGB 色彩模型

RGB 颜色模型广泛用于彩色监视器和计算机，它以红、绿、蓝三种颜色为原色，图 a 呈现了三原色叠加的效果，图 b 所示为该模型所考虑的彩色子空间，其中 X 轴、Y 轴和 Z 轴分别表示红色、绿色和蓝色，三种原色的值分别居于 3 个角上，黑色居于原点，白色则处于距离原点最远的角上



- CIE Lab 色彩模型

CIE Lab 色彩模型在理论上具有最宽的色域，是一种用于非自照明的色彩模型。CIE Lab 色彩模型与设备以及光线均无关，它是由 CIE XYZ 色彩模型经简单变换后得到的，其具有视觉统一性。CIE Lab 颜色模型中 L^* 表示物体亮度； a^* 表示色度，它的最小负值表示绿色，最大正值则表示红色； b^* 也表示色度，它的最小负值表示蓝色，最大正值则表示黄色。

2. 数学模型与算法

2.1 基于 RGB 色彩空间的线性投影灰度化算法

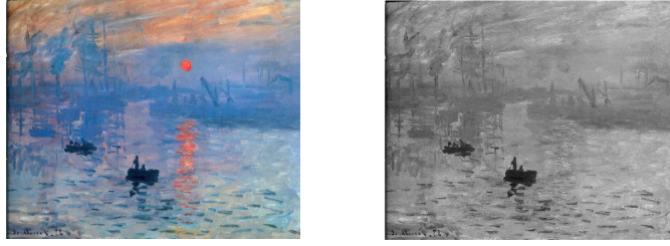
线性投影方法具有快捷、简单和高效的优点，能够在一定程度上保留图像的整体信息，其主要思想是：在 RGB 色彩空间当中寻找一个最佳线性投影方向，把彩色图像的三维像素矩阵沿该最优投影方向转换成一个二维灰度矩阵，完成彩色图像到灰度图像的转换。现如今有许多不同的彩色图像灰度化算法，其中主要包含固定权重的线性投影灰度化算法和动态权重的线性投影灰度化算法。传统的固定权重线性投影灰度化算法主要有 Intensity 算法、Luminance 算法 Luma 算法以及直接取最大值方法。

- Intensity 算法

Intensity 算法是一种平均法，是将彩色图像的三个分量均以 $1/3$ 的权重映射到灰度图像，即

$$R^* = G^* = B^* = (R + G + B)/3 = 0.333R + 0.333G + 0.333B \quad (1)$$

该算法的实验结果如下，从图中可看出该算法能较好的保持图像整体结构信息，但灰度图像却存在严重失真的问题。引起灰度图像严重失真是因为 **R**、**G**、**B** 三个分量总和相同的像素经过该算法都会被映射到同样的灰度级上，因此经该算法得到的灰度图像的可能颜色取值只有 766 种，而原图像的可能颜色取值却高达 3^{256} 种。

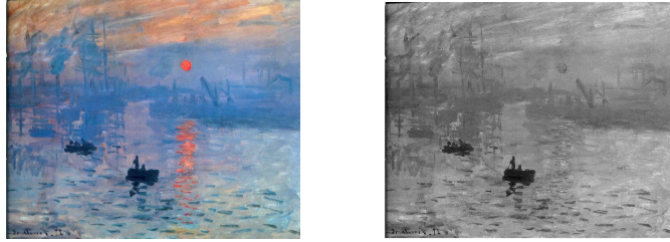


- **Luma 算法**

Luma 算法是一种加权平均法，具体公式如下：

$$gray(x, y) = 0.2126 * R^*(x, y) + 0.7152 * G^*(x, y) + 0.0772 * B^*(x, y) \quad (2)$$

公式 (2) 中， $R^*(x, y)$, $G^*(x, y)$, $B^*(x, y)$ 分别为输入彩色图像经过伽马校正后的分量， $\gamma = 1/2.2$ 。而在本实验中，我们直接取 R, G, B 分量加权，不进行伽马校正。**Luma** 算法的实验结果如图所示，经该算法得出的灰度图像在视觉效果上有一定程度的变亮，但在一些关键信息的表现（如太阳）仍不够好。

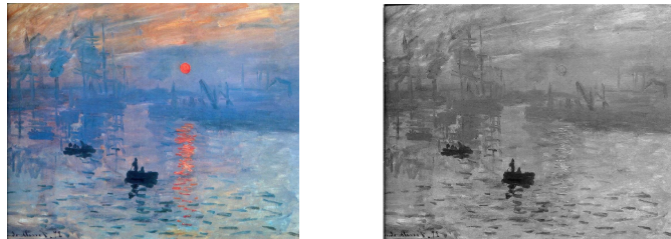


- **Luminance 算法**

Luminance 算法是经典的加权平均法，是依据心理学公式转换得到的结果。该算法被广泛应用在数字图像处理软件中，例如图像编辑软件 **Matlab** 中的 **rgb2gray** 函数，具体公式如下：

$$gray(x, y) = 0.299 * R(x, y) + 0.587 * G(x, y) + 0.144 * B(x, y) \quad (3)$$

结果如下，可以看出太阳和其背景被映射到几乎同一灰度级，效果不太理想。

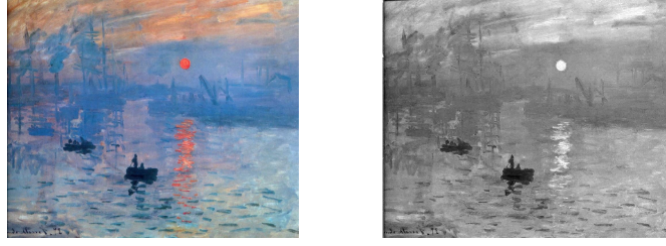


- **直接取最大值方法**

该方法采取的灰度化策略为：

$$gray(x, y) = \max\{R, G, B\} \quad (4)$$

可以看出最大值方法在该图上的测试效果较好，这是因为图片中太阳的R分量较大，直接取最大值有利于太阳从中较为明显地区别出来，但对于其它图片该方法的效果可能就会不太理想。



- 动态权重的线性投影灰度化算法

动态权重的线性投影灰度化算法是通过动态地对彩色图像信息进行评估，然后选取一个最优线性投影方向。2016年，Wan等人提出了一种基于图像信息熵最大化的彩色图像灰度化算法，该算法首先寻找一个最优的投影方向，使得按照该最优投影方向得到的灰度图像具备最大的图像信息熵，然后再通过直方图规格化对灰度图的对比度进行进一步的增强。算法具体流程如下：

首先，通过离散搜索的策略来确定最优投影方向。离散权重 $\omega = (\omega_r, \omega_g, \omega_b)$ ，离散区间为 $[0, 1]$ ，离散步长为 0.1 ，并且约束 $\omega_r + \omega_g + \omega_b = 1$ ，按此方式离散三通道的权重，可以得到 66 种不同权重系数的组合，生成 66 个备选灰度图，对于每一个权重 ω_m 有：

$$gray_m(i, j) = \omega_{r,m} R(i, j) + \omega_{g,m} G(i, j) + \omega_{b,m} B(i, j) \quad (5)$$

针对每一个备选灰度图，计算其图像信息熵：

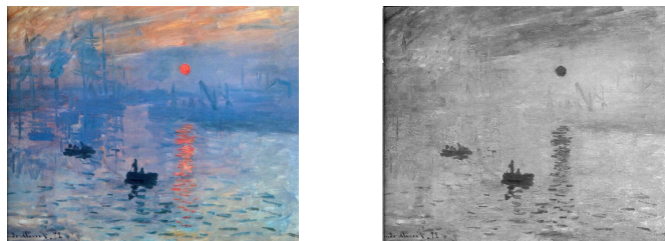
$$H_m = \sum_{k=0}^{255} -(p_{m,k}) * \log_2 p_{m,k} \quad (6)$$

公式 (6) 中， $p_{m,k}$ 表示权重为 ω_m 的备选灰度图当中第 k 灰度级出现的频率，可表示为：

$$p_{m,k} = N_{m,k} / N \quad (7)$$

从 66 个备选的灰度图当中找出图像信息熵最大的灰度图，并将对应的权重定义为 ω_m^* 。把 ω_m^* 当成最佳线性投影方向的近似解，得到相应的最佳投影灰度图。然后，再把最佳投影灰度图进行直方图规格化处理，得出最终的灰度化结果。

通过实验结果可以看出，该算法可以很好的提取原彩色图像的信息，但在寻找最佳线性投影方向时仅仅考虑了备选灰度图的图像信息熵，难以保证输出灰度图像与彩色图像的相似度问题。



通过对上述算法的介绍和实验，可以知道固定权重的线性投影算法没有提取结构信息和对比度信息的能力，具有很大的局限性；动态权重的线性投影灰度化算法通常采用离散搜索的方式，在有限个备选灰度图中选择一个最优灰度图，但该最优灰度图并不是严格意义上的最优灰度图，只是最优灰度图的近似解，所以往往存在特征信息丢失的情况。但是，线性投影灰度化算法都具有简单、高效和快捷的优点。

2.2 实时对比度保留的图像去色算法 (RTCP)

Lu 等人在文献中提出了一种实时对比保存脱色算法，具体算法如下：

首先, 定义映射函数为 $g = f(c)$, 对于每个输入 **RGB** 矢量 $c = (c_r, c_g, c_b)$, 函数 f 的输出 g 为相应的灰度值。算法采用全局映射方案, 具有相同颜色的像素被转换为相同的灰度级, 并采用有限多元多项式函数定义颜色 c 的多项式空间:

$$g = \omega_1 c_r + \omega_2 c_g + \omega_3 c_b + \omega_4 c_{rg} + \omega_5 c_{rb} + \omega_6 c_{gb} + \omega_7 c_r^2 + \omega_8 c_g^2 + \omega_9 c_b^2 \quad (8)$$

其次, 通过最小化能量函数来估计像素 x 和 y 的灰度级:

$$\min_g \sum_{(x,y) \in p} (g_x - g_y - \delta_{x,y})^2 \quad (9)$$

式 (8) 中 g_x 和 g_y 分别表示像素 x 和 y 的灰度级, $\delta_{x,y}$ 为颜色对比度, 在 **CIELab** 颜色空间中, 可以表示为:

$$|\delta_{x,y}| = \sqrt{(L_x - L_y)^2 + (a_x - a_y)^2 + (b_x - b_y)^2} \quad (10)$$

定义弱色彩顺序约束:

$$\alpha_{x,y} = \begin{cases} 1 & \text{if } r_x \leq r_y, g_x \leq g_y, b_x \leq b_y \\ \text{or } r_x \geq r_y, g_x \geq g_y, b_x \geq b_y \\ 0.5 & \text{otherwise} \end{cases} \quad (11)$$

如果 $\alpha_{x,y} = 1$, 将先验 $G(\delta_{x,y}, \sigma^2)$ 应用于明确的颜色顺序执行, 否则让颜色差异遵循双峰分布, 因此 p 中像素对的最终能量函数表示为:

$$\prod_{(x,y) \in p} \{ \alpha_{x,y} G(\delta_{x,y}, \sigma^2) + (1 - \alpha_{x,y}) G(-\delta_{x,y}, \sigma^2) \} \quad (12)$$

最大化公式(2-27)相当于最小化它的负对数:

$$E_p(g) = - \sum_{(x,y) \in p} \ln \in \{ \alpha_{x,y} G(\delta_{x,y}, \sigma^2) + (1 - \alpha_{x,y}) G(-\delta_{x,y}, \sigma^2) \} \quad (13)$$

其中 $G(\delta_{x,y}, \sigma^2) = \exp\{-\frac{(g_x - g_y + \delta_{x,y})^2}{2\sigma^2}\}$

至此, 得到了最终需要优化的能量函数。

为了获取 g 的值, 我们采用最简单, 最常用的方法, 通过对图像像素的**RGB**统一加权获得。只不过, 这一权重是通过优化上述能量函数得到, 而不是简单定义。即需要通过优化来获得权重 $\omega_r, \omega_g, \omega_b$, 满足:

$$\begin{aligned} \omega_r &\geq 0, \omega_g \geq 0, \omega_b \geq 0 \\ \omega_r + \omega_g + \omega_b &= 1 \end{aligned} \quad (14)$$

为了简化计算, 我们将连续的优化问题转换为离散的, 不妨认为三个颜色通道分别的颜色系数是在{0, 0.1, 0.2, ..., 1.0}这11个值中间选取的, 这样系数的不同选取只有66种不同的选择。此外, 考虑到一般图像分辨率较高, 通常存在内在冗余, 我们可以将图像降采样为 64×64 的分辨率 (实际编程时降为 128×128 以下即停止)。



2.3 PCA

主成分分析算法（PCA），主要通过计算选择特征值较大的特征向量来对原始数据进行线性变换达到降维的目的。应用主成分分析到图像灰度化，主要的算法流程如下：

输入：图片数据 image

1. 对所有样本中心化： $x_i \leftarrow x_i - \frac{1}{n} \sum_{i=1}^n x_i$
2. 计算样本的协方差矩阵： XX^T
3. 对协方差矩阵做特征值分解
4. 取协方差矩阵的最大特征值以及其所对应的特征向量，作为数据集的主方向（可能需要反向操作），将原数据向主方向上投影
5. 对新数据作线性映射，使其对应到原数据集的范围

输出：灰度化后的图片 image_gray

实验结果如下，可以看到PCA 的效果还是很好的，把太阳从背景中较为明显地凸显出来，且对于其它的轮廓也展示的较为清晰。



2.4 基于主成分分析的新型灰度化算法

2013 年 Seo 等人提出了一种新的基于主成分分析(PCA)的彩色转灰度算法。首先把原始彩色图像从 RGB 颜色模型变换至 YCbCr 颜色模型；然后在 YCbCr 颜色模型中对输入图像进行 PCA 分析降维得到三个特征向量，其中能量最高的特征向量为投影的主要方向。算法具体实现过程如下：

首先，把原始彩色图像从 RGB 颜色模型变换至 YCbCr 颜色模型，并去掉直流分量：

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256} \cdot \begin{bmatrix} 65.738 & 129.057 & 25.06 \\ -37.945 & -74.494 & 112.43 \\ 112.439 & -94.154 & -18.28 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (15)$$
$$I_{ycc} = I_{ycc} - \bar{I}_{ycc}$$

公式 (2-19) 中, \bar{I}_{ycc} 代表 I_{ycc} 的均值。

其次, 对图形 I_{ycc} 进行主成分分析, 得到三个特征向量 $v_i (i = 1, 2, 3)$ 和对应的三个特征值 $\lambda_i (i = 1, 2, 3)$, 且 $\lambda_1 \geq \lambda_2 \geq \lambda_3$, 并对特征向量和特征值进行归一化处理:

$$\lambda_i = \lambda_i / \|\lambda\|, v_i = v_i / \|v_i\| \quad (16)$$

然后，线性加权得到初始灰度图:

$$I_{gray} = \sum_{i=1}^3 \lambda_i (v_i^T I_{ycc}) \quad (17)$$

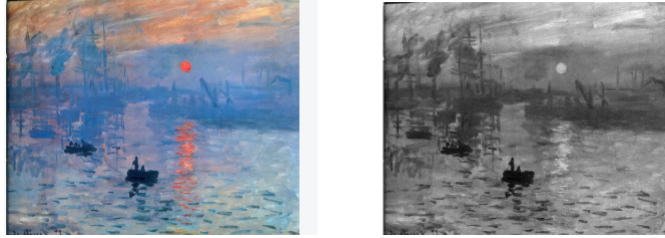
再将初始灰度图数据映射到 $[0, 255]$,

最后, 将满足 $|I_y - I_{\text{gray}}| > |I_y + I_{\text{gray}} - 255|$ 的像素进行灰度反转:

$$I_{\text{gray}}^* = 255 - I_{\text{gray}} \quad (18)$$

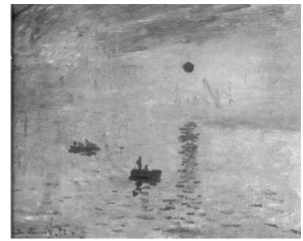
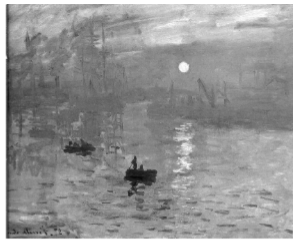
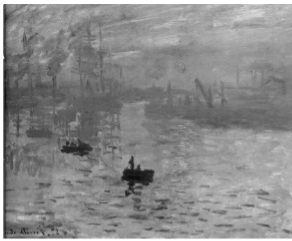
公式 (18) 中的 I_{gray}^* 即为最终的灰度图像。

实验结果如下, 可以看出图像的边缘轮廓信息和细节信息保留的比较完整, 但是在色彩还原方面还具有
一些瑕疵



3. 实验结果

在该部分中, 我们展示 **Luminance** 算法, 最大值法, 动态权重的线性投影法, **RTCP**方法, **PCA**和基于 **PCA**的新型算法这六种方法在测试图片上的结果:



4. 结论

从上面的实验结果可以看出，PCA法和基于PCA的新型算法较好地保留了原始图像的结构信息及局部对比度信息，并且普适性更好；线性投影化算法的灰度化效果不太理想，丢失了较多原图像的对比信息，可能会把两种颜色相差较大的物体几乎映射到同一灰度级上，灰度化效果不稳定，但其也具有简单便捷、运算效率高的优点。

参考文献

- [1] 刘美.彩色图像灰度化算法的研究与实现[C].长春：长春理工大学，2019-06
- [2] Seo J W, Kim S D. Novel PCA-based color-to-gray image conversion[C]. IEEE International Conference on Image Processing, 2013: 2279-2283.
- [3] Cewu Lu, Li Xu, and Jiaya Jia. 2012. Contrast Preserving Decolorization. In IEEE ICCP