图像处理作业 1 —— 直方图处理

本次作业中实现了直方图均衡化和直方图规定化的算法.

1 直方图均衡化

用 r 表示待处理图像连续的灰度, $r \in [0, L-1]$, $p_r(r)$ 表示 r 的概率密度, 直方图均衡化的目标是找到一个变换

$$T : [0, L-1] \longrightarrow [0, L-1]$$
$$r \longmapsto s$$

使得 $p_s(s) = \dfrac{1}{L-1}, 0 \leq s \leq L-1$. 可求得

$$T(r) = (L-1) \int_0^r p_r(w) \mathrm{d}w$$

改写成离散形式

$$s_k = T^*(r_k) = (L-1) \sum_{j=0}^{k} p_r(r_j), \quad k = 0, 1, \cdots, L-1$$

因此, 对于给定的图片, 只需用`cv2.calcHist()`函数求出直方图后, 再利用`numpy.cumsum()`进行累加操作, 即可完成直方图均衡化, 均衡化结果如 fig. 1

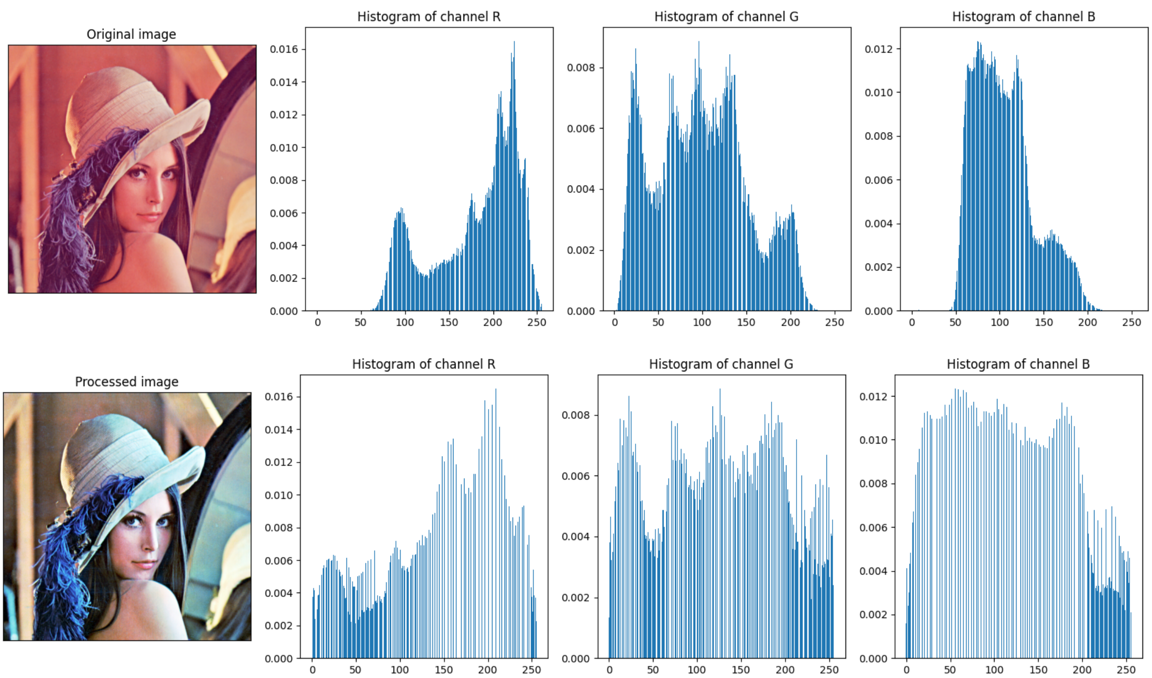

图 1: 直方图均衡化结果. 上下方分别为原图和均衡化后的图片及三通道直方图

2 直方图规定化

对于灰度 $r \in [0, L-1]$ 和概率密度 $p_r(r)$, 给定一个新的随机变量 $z \in [0, L-1]$ 和概率密度 $p_z(z)$, 直方图规定化的目标是找到一个变换

$$\begin{aligned} T : [0, L-1] &\longrightarrow [0, L-1] \\ r &\longmapsto s \end{aligned}$$

使得 $p_s(s) \equiv p_z(z)$, 理论上可以先求出 r 和 z 的均衡化变换 T_r 和 T_z, 从而得到 $T = T_z^{-1} \circ T_r$, 但有时 T_z^{-1} 并不严格存在 (映射不唯一), 即使对于离散后的情况 (用 * 表示一个变换的离散化), 也需要另外做规定, 才能真正确定 T^*. 因此, 在实现过程中, 采取另一种较为简洁的方式, 并不直接求 T_r^* 和 T_z^*, 而是考虑 $\sum\limits_i p_r(r_i)$ 和 $\sum\limits_i p_z(z_i)$ 的关系.

$$\sum_{i=0}^{k} p_r(r_i) \approx \sum_{i=0}^{T^*(k)} p_z(z_i) \tag{1}$$

记 $S_{r,k} = \sum\limits_{i=0}^{k} p_r(r_i)$ 和 $S_{z,k} = \sum\limits_{i=0}^{k} p_z(z_i)$, 取一个小量 $\varepsilon > 0$, 构造定义在 $[0, 1]$ 上的分段线性, 且严格单调递增的函数 F, 其所有折点为

$$F(S_{z,k} + k\varepsilon) = k, \quad k = 0, 1, \cdots, L-1$$

则对于任意 $k \in [0, L-1]$, 存在一个 k', 使得

$$\begin{aligned} k' = F(S_{z,k'} + k'\varepsilon) &\approx F(S_{z,k'}) \\ &\leq F(S_{r,k}) \\ &\leq F(S_{z,k'+1}) \approx F(S_{z,k'+1} + (k'+1)\varepsilon) = k' + 1 \end{aligned}$$

由于 F 是严格递增的, 故 k' 或 $k'+1$ 最能够代替 eq. (1) 的 $T^*(k)$. 因此可以取

$$T^*(k) = \mathrm{round}(F(S_{r,k}))$$

其中 $S_{r,k}$ 可以利用`numpy.cumsum()`进行计算, 而函数 $F(\cdot)$ 也能由`numpy.interp()`立刻构造得到.

直方图规定化的结果见 fig. 2

3 精确直方图规定化

一般的直方图规定化和均衡化总会有离散值、近似等因素带来的误差, 如 fig. 1中均衡化后的直方图并非处处相等, fig. 2中规定化后的直方图也与给定的直方图不同. 精确直方图规

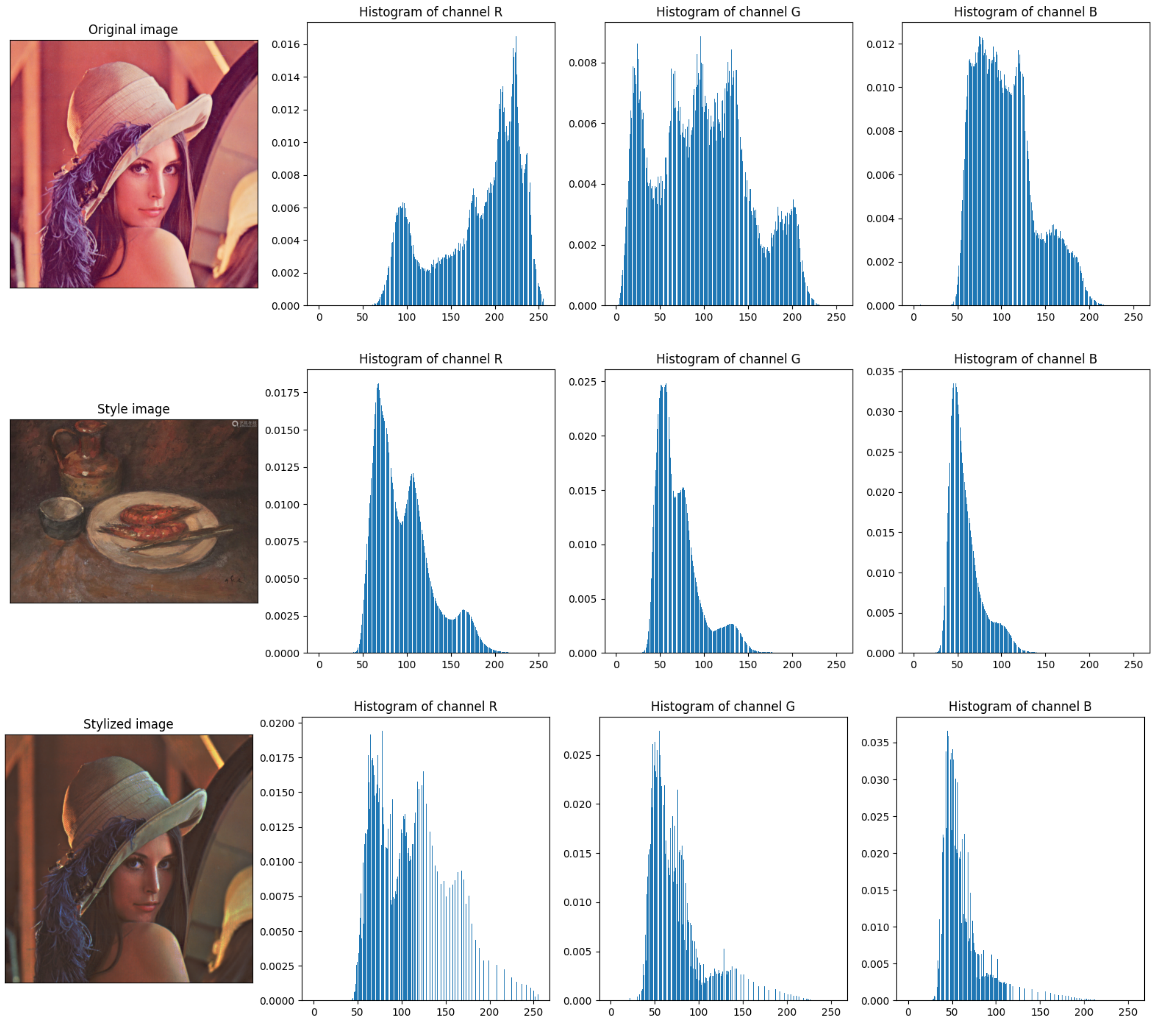

图 2: 直方图规定化结果. 自上至下分别为原图、目标直方图的来源、规定化后的图片及它们的三通道直方图

定化的目的便是尽量消除这些误差, 一个有效的算法是 [1], 通过对像素进行排序、分组, 并引入一系列 filters 作用于图像.

在本次作业中实现的算法则基于更为朴素的想法, 通过逐颜色比较原图的直方图 $\{p_i\}_{i=0}^{L-1}$ 和目标直方图 $\{q_i\}_{i=0}^{L-1}$, 假设图片大小为 $M \times N$, 让 i 遍历 $0, 1, \cdots, L-1$

- 若 $p_i > q_i$, 则任意取出 $MN(p_i - q_i)$ 个颜色为 i 的像素, 称之为 "多余像素"
- 若 $p_i < q_i$, 则将 $MN(q_i - p_i)$ 个 "多余像素" 的颜色变为 i

可以用一个队列Q记录需要改变颜色的像素位置, 可以使得颜色值较小的多余像素转换的目标颜色也小, 从而尽可能保持新图与原图的一致性.

精确规定化的结果见 fig. 3, 采取了两种方法进行精确规定化: 一是对原图直接精确规定化; 二是先进行规定化, 在进行精确规定化. 从图中可以看出, 对原图直接精确规定化后的结果噪声极大且色彩分布情况混乱.

下面定性分析这一现象, 以人脸、帽子和背后的墙壁在规定化后图片中呈现绿色为例: 在原图中这部分区域既存在 R 值较大而 G 值稍小; 而目标直方图在 R 值大于 200 的区间分布接近 0, 因此这些区域较大的 R 值成为 “多余像素” 而变为一个极小的值, 致使未改变的 G 成为值最大的通道而呈现绿色.

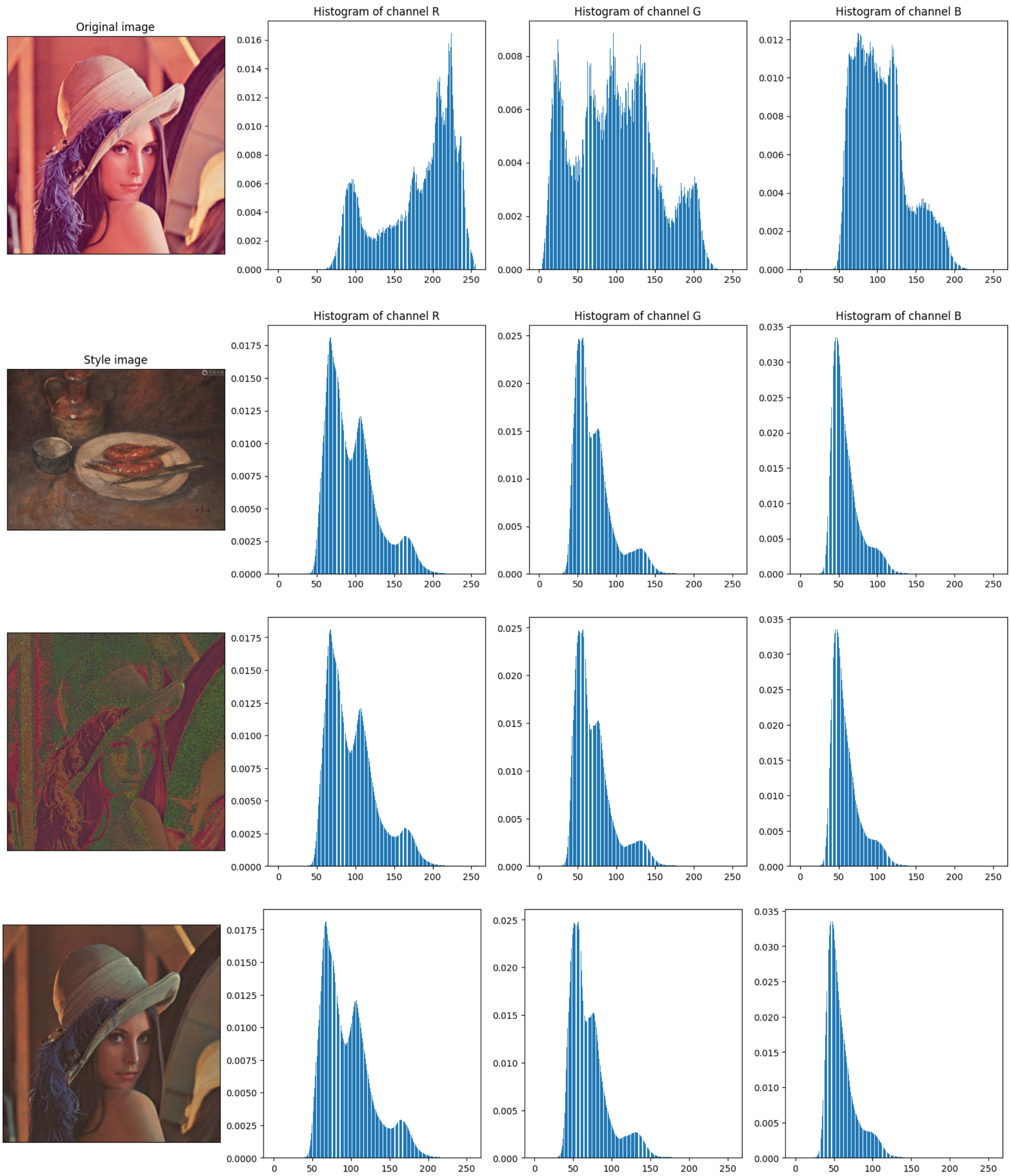

图 3: 精确直方图规定化结果. 自上至下分别为原图、目标直方图的来源、直接精确规定化后的图片、先进行规定化再进行精确规定化的图片, 及它们的三通道直方图

4 利用直方图规定化进行信息加密

本部分灵感来源于1st PKU GeekGame中的题目《叶子的新歌》.

加密信息的基本思路是用将一个有意义的波形作为直方图 (如条形码), 再选择一张图片对其进行精确直方图规定化.

fig. 4展示了利用直方图规定化对条形码进行加密的结果.

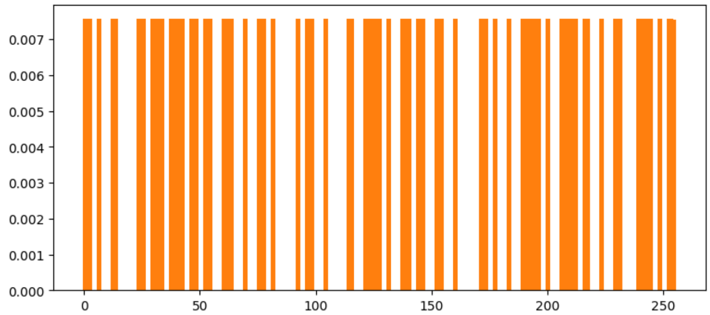

图 4: 直方图加密. 右侧为左图 R 通道的直方图.

参考文献

[1] Dinu Coltuc, Philippe Bolon, and J-M Chassery. Exact histogram specification. *IEEE Transactions on Image processing*, 15(5):1143–1152, 2006.