多媒体技术Project2 说明文档

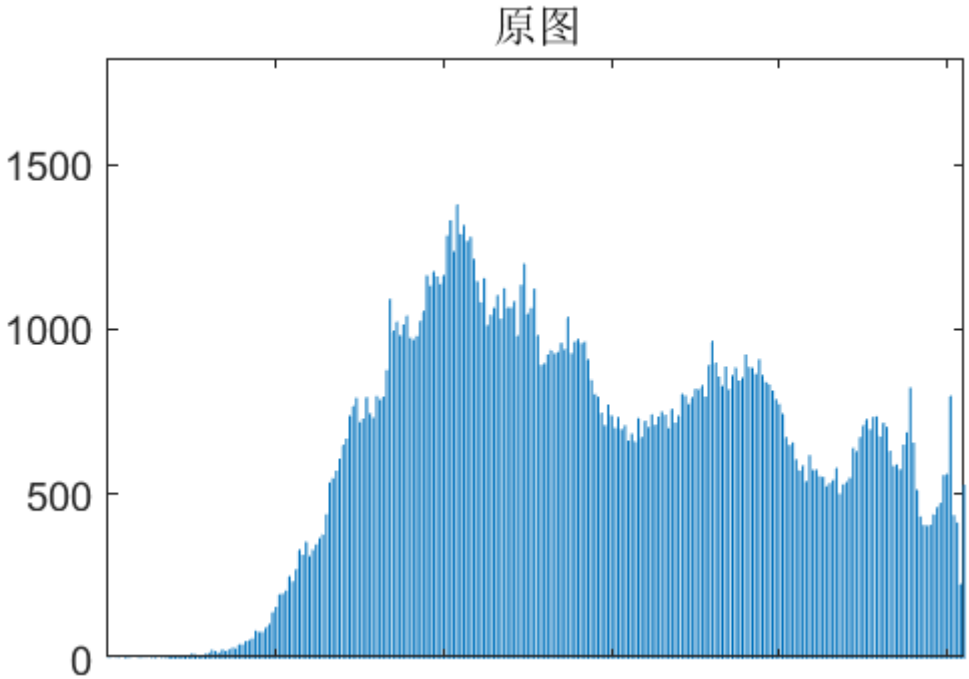
18302010013 王中亮 软件工程

第三题

在均衡化过程中，像素首先应保持原来的大小关系不变，即明暗不会变化，只是增大对比度，另外，像素映射的值域不能越界。

以下图片中，均以此图为原图进行说明：

其对应的灰度分布如下：



直方图均衡化

直方图均衡化的作用是图像增强，根据上述条件，选用的是累积分布函数。

$$S_k = \Sigma_{j=0}^{k} \frac{n_j}{n} (k = 0, 1, 2, \ldots, L-1)$$

n = 图像中像素的总和，n_k = 当前灰度级的像素个数，L = 图像中可能灰度级总数

累积分布函数单调递增且值域在0-1符合要求，一副图像的灰度级可以看成区间[0,L-1]的随机变量。

直方图均衡化的做法是将原来的灰度值通过求和的方式获得新的灰度值。

从公式来看，经过操作之后的灰度之差，从原来的

$$n_{k+1} - n_k \text{变为了} \frac{n_{k+1}}{n}$$

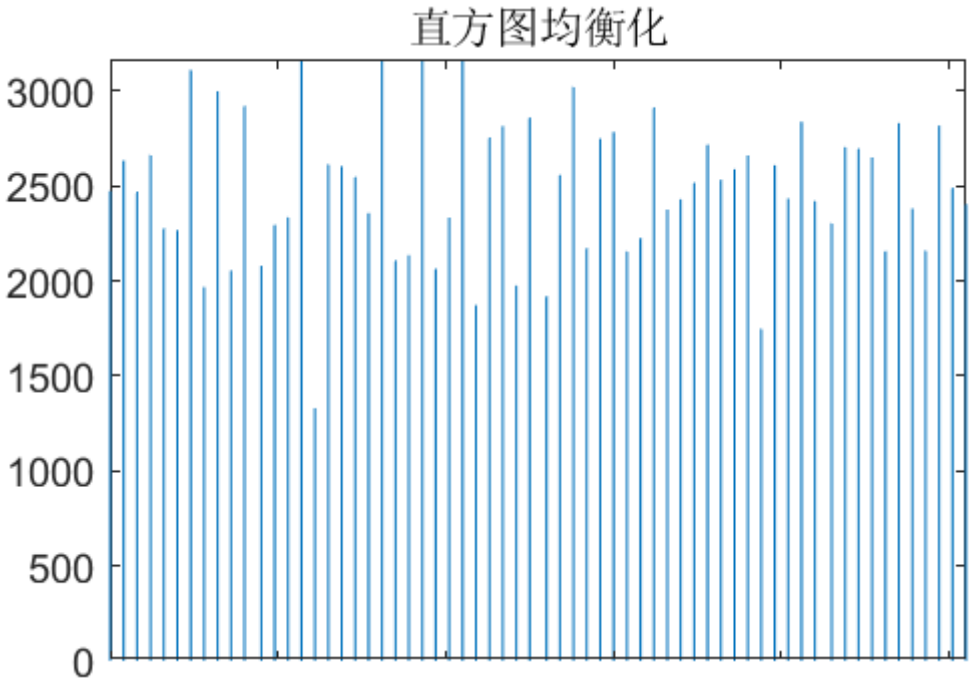
显然此时的灰度之差的变化取决于原本的图像的灰度分布，对于原始差距较大的地方来说，差距略微缩小了，而较小的地方则略微增大。

$$n(n_{k+1} - n_k) ? n_{k+1}$$

$$(n-1)n_{k+1} ? n * n_k$$

$$\frac{n_{k+1}}{n_k} ? \frac{n}{n-1}$$

？揭示了灰度之差的变化，而显然当原始之差更大时，？应取 >，意味着这个差距略微缩小了。



从灰度分布直方图来看，也能明显看出，在经过直方图均衡化之后，灰度分布变得很均匀。

灰度拉伸

而灰度拉伸时通过对对比度拉伸来达到增强对比度的效果

$$F(x,y)=\frac{f(x,y)-MIN_{f(x,y)}}{MAX_{f(x,y)}-MIN_{f(x,y)}}$$

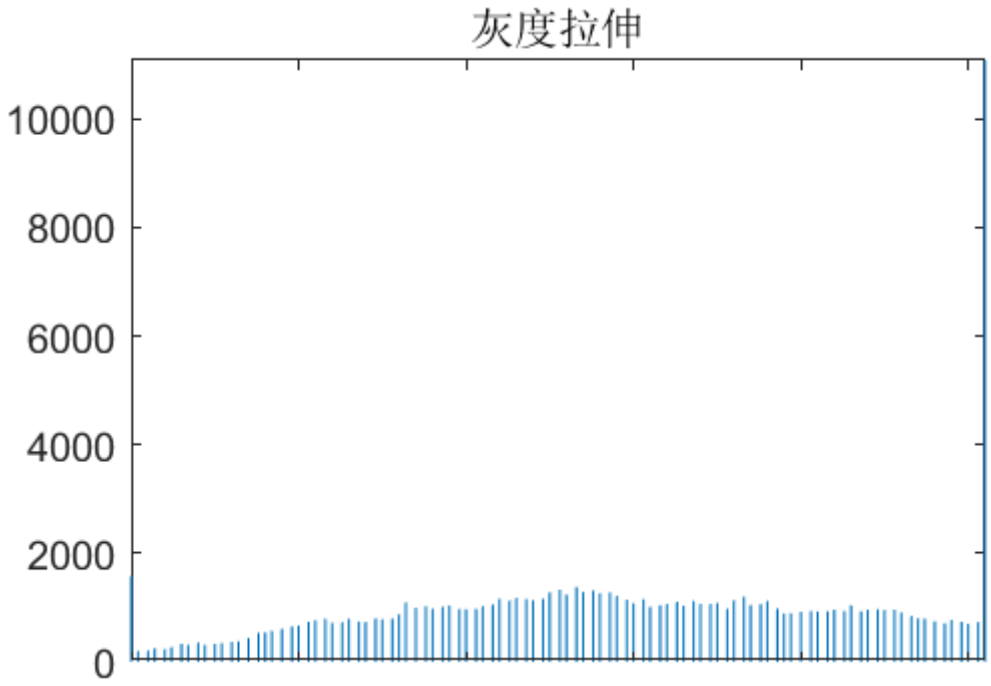
可以看出，灰度拉伸的函数会将原来的灰度之差增大，其倍数即为原有的对比度大小，由于灰度差距的，原来较亮的地方显得更亮，更暗的地方显得更暗。简单的来说，就是将原本的灰度区间强行拉伸到了[0,1]这一范围。

对于对比度较小的图片来说，会使得图片失真明显。

例如：

在灰度最大的地方，由于根据公式，其灰度值为1，即最大值，会成全黑，而在灰度最小的地方，直接变成了0。

在灰度值为0.3的地方，假设灰度最高为0.7，最低为0.1，经过函数代换得该点的灰度为0.33，显得更暗，在0.2的位置，则变成0.18，显得更亮。



从灰度拉伸的灰度分布表也能轻易看出，其图像分布和原图的灰度分布较为接近，但数值来看，高低差变大了许多。

第四题

原图如下：

经过椒盐增噪之后的图片如下：

加噪声

椒盐噪声是一种随机出现的噪点，由于其产生的原因常常是传输设备/影像讯号收到突如其来的强烈干扰导致，因此其分布应不属于正态分布等常见分布中的任何一种

方法一：高斯滤波

$$h(X) = k_d^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\varepsilon) c(\varepsilon, X) d\varepsilon$$

$$h(X) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\varepsilon, X) d\varepsilon \text{为归一化函数，保证权重之和等于} 1$$

高斯滤波是通过一个用户指定的模板来扫描图像中每一个像素，用模板确定的邻域内的像素加权平均灰度值去代替模板中心像素的点，其加权即为该点到中心点的距离。

结果如下：

高斯滤波

高斯滤波会使得图像变得**模糊。**由于在分析各个点（即卷积）的过程中，每个像素点都取周围像素的平均值，在数值上是一种平滑化，类似于之前提到的直方图均衡化的处理。这种处理在图像上就表现为**模糊**，中间点失去细节。

对于不服从正态分布的噪音，例如本例中的椒盐噪音，高斯模糊显得非常乏力，由于在特定的噪点分布的位置，噪点的权重最大，高斯滤波实际上可以看作是将噪点给抹开了变得均匀一些，甚至可以说这个抹开的过程也是尽可能地将噪点均匀抹在整张图片上，并不能做到消除噪点。

方法二：均值滤波

均值滤波和高斯滤波很相似，区别在于，均值滤波主要是**用像素点周围的8个点来构成一个模板，再用模板中的点来取代原来的像素值**，这个模板和高斯滤波的加权一样，也可以进行调整。

平均模板滤波

平均模板滤波也会使得图片变得**模糊**，相比于高斯滤波更加严重。主要原因在于，处于同一个模板内的点的像素值过于接近了，对于相邻的两个点来说，他们的取样差别仅仅只有最左侧和最右侧的3个像素（对于模板是周围8个像素和自身构成的9像素模板），经过平均之后差别不大，而高斯滤波每个点都需要考虑其再整张图的位置，扩大了平均之后的差别。

对于噪点的处理来看，平均模板滤波却切实的做到了让噪点的深度变淡，但和高斯分布相似，也将噪点抹开了。主要区别在于，平均模板分布不会尽可能地抹开像素点，而是仅仅会影响周围8个格子，且在经过平均模板滤波之后噪点的像素值会明显减弱。

方法三：中值滤波

中值滤波是非线性的滤波，因此和高斯滤波和均值滤波无论是原理上还是表现上都有很大的区别。对于一个数字信号序列 **x_j** 进行处理的场合，首先定义一个长度为

$$L = 2N + 1 (N_{为整数})$$

的长窗口，在某一时刻，窗口内的信号样本定义为

$$x(i-N), \ldots, x(i), \ldots, x(i+N) (i_{表示处于中间的数})$$

对这L个信号进行升序排序后，其中值滤波的输出值如下：

$$y(i) = Med[x(i-N)], \ldots, x(i), \ldots, x(i+N)$$

中值滤波是在 **“最小绝对误差”** 准则下的最优滤波

中值滤波

中值滤波的缺点在于，窗口较大时，滤波的计算量会迅速提升，因此需要寻找一个最适合中值滤波的算法来进行操作。此外，MATLAB的中值滤波操作只支持二维数组，需要将一张RGB图片的三个通道分开分别进行降噪处理之后合并，这一过程也会存在损耗。

由于中值滤波是非线性的，因此在保护边缘信息和保留细节方面较为优秀，但从输出的图片来看，人物的边缘勾线也存在一些坑洼，这是不可避免的。

中值滤波对于消除椒盐噪声非常有效，因为椒盐噪声在长窗口中，几乎不存在排序至中值的情况（倘若椒盐噪声的噪点能够排序至中值，意味着这个噪点几乎和图片融合在了一起，那么其是否能够称之为噪点还是一个值得讨论的命题）。从这个角度来看，中值滤波在处理椒盐噪声的情形下无疑是最优解。