

计算机视觉作业二

姓名：李宁

学号：2120180459

专业：计算机科学与技术

问题一

一、问题描述

给定一张图像 HW2Pic1.png。

- 计算图像的灰度直方图；
- 对该图像做直方图均衡化，是图像亮度看着更好。

二、基本思路

通过对亮度和增益的控制可以改善图像的显示，通常有两种方法：一种方法是寻找图像中最亮和最暗的像素值，将它们映射到纯白和纯黑；另一种方法是寻找图像中像素的平均值作为中间灰度值，然后扩展范围以达到尽量充满可显示的值。这里我们采用第一种方法，即直方图均衡化。

对于图像的直方图均衡化，即寻找一个映射函数 $f(I)$ ，图像的各个通道的直方图经过映射后是平坦的。寻找此映射的方法与从概率密度分布函数产生随机样本的方法类似，其中主要是计算灰度图像像素分布的累积分布函数。

假设一个原始的直方图 $h(I)$ 为一张灰度图像中的像素分布，则可以通过 $h(I)$ 的分布得到累计分布函数 $c(I)$ ：

$$c(I) = \frac{1}{N} \sum_{i=0}^I h(i) = c(I-1) + \frac{1}{N} h(I)$$

其中 N 是图像中像素的总个数。对于任意给定的亮度，我们可以查出它对应的百分比 $c(I)$ ，此时可以决定此像素所对应的最终的值。当对 8 位的图像的值进行操作时，坐标轴 I 和 c 要缩放到 $(0, 255)$ 。

三、原理及算法

在解决该问题过程中，其核心便是的到输入图像的像素分布的累积分布函数，然后根据原始像素的累积分布函数将其重新映射到 $(0, 255)$ 。具体分以下几个步骤进行：

- ① 首先，将输入图像转换为灰度图

```
im_gray <— im_source
```

- ② 获取灰度图像的灰度直方图以及累积直方图

```
im_hist <— histogram(im_gray)
```

```
cdh <— cumsum(im_hist)    #cdh 为 1*256 矢量，存储  $(0, 255)$  对应像素
```

③ 分离 RGB 图像的各个通道，对每个通道的像素计算 0 到 255 的累积分布函数映射

```
im_R <— im_source[0]
```

```
im_G <— im_source[1]
```

```
im_B <— im_source[2]
```

```
im_r[i] <— cdh[im_R[i]] * 255 / N #对原像素第 i 个位置进行累积分布函数映射
```

```
im_g[i] <— cdh[im_G[i]] * 255 / N
```

```
im_b[i] <— cdh[im_B[i]] * 255 / N
```

④ 合并三个颜色通道，获得最终处理后的图像

```
im_target <— Combine(im_r, im_g, im_b)
```

四、实验结果与分析

1、最终获得的实验结果如下：



图 1 原始图像（左）与直方图均衡化图像（右）

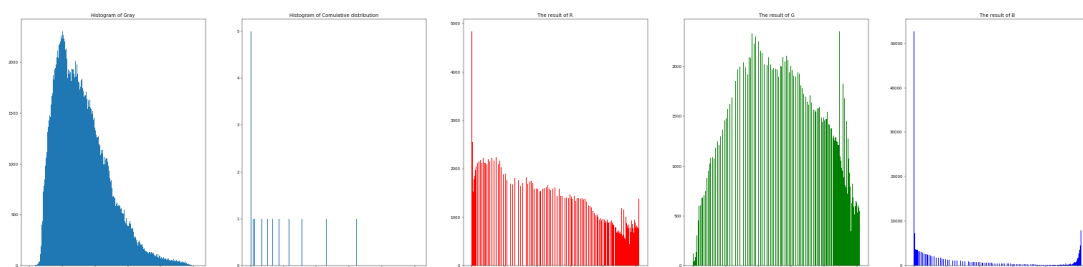


图 2 由左到右：灰度图直方图、灰度累积直方图、均衡化后的 R，G，B 通道分布直方图

2、结果分析

如上图 2，在直方图均衡化之前，图像像素点整体的分布更偏向于像素值小的像素点。经过直方图均衡化之后，图像的 R，G，B 三个通道上像素点的分布较之前较为“平坦”。

但是，在该实验实现过程中还存在着如下缺陷：

- 简单的对各个颜色通道进行直方图均衡化处理在一定程度上会影响图像的色度和色调。该问题可以通过提取原始图像的亮度图像，利用颜色比值来进行处理。
- 通过 R、G、B 三个通道直方图均衡化之后的结果，处理后的直方图看上去并不平滑，个人认为可以对每一张直方图采用局部补偿的方法改善其效果。

（补偿函数形式为： $f(I) = \alpha c(I) + (1 - \alpha)I$ ）

问题二

一、问题描述

给定一张图象 HW2Pic2.png，完成如下工作。

- 对图像叠加高斯噪声，噪声分布满足 $N(0,0.01I)$ ；
- 对噪声图像执行高斯滤波、中值滤波和双边滤波；
- 对三种滤波器的处理结果进行分析

二、基本思路

1、高斯噪声的叠加

由于输入图像是 RGB 彩色图像，所以需要原始图像的每一个颜色通道上的像素进行叠加高斯噪声。

因此首先要将原始的 R、G、B 三个通道分别分离出来，将随机生成的一个高斯噪声值叠加到每个通道的相应像素上去，最后再将三个分通道合并为一张图象，高斯噪声即叠加完成。

2、高斯滤波

高斯滤波器是一种对像素有平滑作用的可分离滤波器。在数值图像处理中，高斯滤波主要可以使用两种方法实现：①使用一种离散化的滑动窗口卷积；②通过傅里叶变化进行操作。在本次实验中，使用的是第一种方法。

关于如何构造一个高斯核以及如何确定每一个参数的值，最核心便是高斯函数公式，对于二维矢量 $[x, y]$ ，当矢量中各分量的均值和方差相同时，一个二元高斯函数可以有列公式表示：

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

其中 x^2 和 y^2 分别表示的是邻域内其他像素与邻域内中心像素的距离， σ 表示的是标准差。

实验过程中并没有对高斯核的构造过程进行实现，为了方便，使用了如下一个已有的高斯核：

$$\frac{1}{256} * \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

由于该滤波核是对称的，所以在与图像局部区域做卷积时是否旋转 180 度均可。

3、中值滤波

与高斯滤波不同，中值滤波是一种邻域非线性滤波操作。当噪声是散粒噪声，而不是高斯噪声，即图像偶尔会出现很大的值，在这种情况下高斯滤波器只是将噪声转换为更柔和但仍然可见的散粒，不会去除噪声像素。在这种情况下中值滤波器是一个较好的选择，它选择每个像素的邻域像素中值作为输出。中值滤波对模糊的处理较好，且保证了输出像素是已经出现过的。

中值滤波的实现相对简单，仍以一个 5*5 的卷积作为像素邻域范围，在实现过程中只要对滑动窗口所覆盖的图像区域的像素值进行排序取中值作为输出即可。

4、双边滤波

双边滤波器将加权滤波器核和更好的一直外点的方案结合起来，其抑制与中心像素的值差别太大的像素，这便是双边滤波的精华所在。与高斯滤波器等平滑滤波器相比，双边滤波器可以保留截越部分（物体边界）。但它有个缺点，慢!!!!

在双边滤波器中，输出像素的值依赖于邻域像素的值的加权组合，

$$g(i,j) = \frac{\sum_{k,l} f(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

其中，权重系数 $w(i,j,k,l)$ 取决于定义域核

$$d(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right)$$

和依赖于数据的值域核

$$r(i,j,k,l) = \exp\left(-\frac{\|f(i,j) - f(k,l)\|^2}{2\sigma_r^2}\right)$$

的乘积。它们相乘后，就会产生依赖于数据的双边权重函数

$$w(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2}\right)$$

值得注意的是，定义域核实质上是普通的高斯核，值域核使用的是中心像素于邻域像素的矢量距离。值域核衡量了邻域像素与中心像素的亮度相似性，这一点对于彩色图像很重要。

5、卷积操作保持尺寸不变

当图像与卷积核进行卷积后图像尺寸会变小，为了使卷积后的图像尺寸不变，这里简单给出如下尺寸变换公式：

$$W(H) = [(输入大小 - 卷积核大小 + 2 * P) / 步长] + 1$$

令变换后的 W 和 H 保持不变，便可求出原图像需要添加 padding（公式中 P）的数量。该实验过程中，卷积核大小为 5*5，步长为 1，因此 P=2。

三、原理及算法

1、叠加高斯噪声

- 分离源图像的三个通道；
- 在每一个通道相应的像素上加上一个随机生成的高斯函数值；
- 对于像素值小于 0 的像素统一置 0，对大于 255 的像素统一置 255；
- 合并添加完高斯噪声的三个通道，返回叠加了高斯噪声的彩色图像。

2、高斯滤波处理

- 定义一个 5*5 的卷积核；
- 三个添加 padding 的二维数组 <—— 分离 RGB 颜色通道
三个数组分别为 im_tmpr, im_tmprg, im_tmprb；
- 对上述三个二维数组，利用滑动窗口，在窗口区域内进行卷积操作；
- 合并 im_tmpr, im_tmprg, im_tmprb；
- 返回高斯滤波后的图像 im_gauss。

3、中值滤波处理

- 定义一个卷积区域（5*5）；
- 三个添加 padding 的二维数组 <—— 分离 RGB 颜色通道
三个数组分别为 im_tmpr, im_tmprg, im_tmprb；
- 对于每一个二维数组，利用滑动窗口，获取窗口范围内像素值的中值作为返回值；
- 合并 im_tmpr, im_tmprg, im_tmprb；
- 返回中值滤波后的图像 im_median；

4、双边滤波处理

- 定义一个卷积区域 (5*5);
- 三个添加 padding 的二维数组 <—— 分离 RGB 颜色通道

三个数组分别为 im_tmpr, im_tmprg, im_tmprb;

- 计算卷积区域定义域核的参数值:

$$d[m][n] <—— d(i, j, k, l) \quad \# m=i+k; n=j+l$$

- 对每一个二维数组, 在滑动窗口区域内进行如下操作:

- ① 计算该区域内值域核的参数值

$$r[m][n] <—— r(i, j, k, l) \quad \# m=i+k; n=j+l$$

- ② 计算该区域每个像素对应的权重参数

$$\text{weight} <—— d * r \quad \# d \text{ 与 } r \text{ 相应元素相乘, weight 为 } m*n \text{ 维}$$

- ③ 计算该区域内的 $g(i, j)$ 函数值作为该区域的返回值, 其中 (i, j) 为中心点坐标。

- 合并 im_tmpr, im_tmprg, im_tmprb;
- 返回双边滤波后的图像 im_blt。

四、实验结果与分析

1、实验结果



图 1 叠加高斯噪声后的图像



图 2 高斯滤波图像（较原图模糊）



图 3 中值滤波图像（边角有黑点）



图 4 双边滤波图像（清晰，超慢）

2、结果分析

针对本次实验，主要分析以下几点。

① 通过高斯滤波、中值滤波、双边滤波对噪声图像的处理结果可以明显看出高斯滤波和中值滤波处理过的图像都在一定程度变得模糊，而经过双边滤波处理的图像较好地保留了图像中物体的边缘，显得较为清晰。

而前两者处理后图像模糊的原因并不同。高斯滤波是由于对图像的每一个像素进行了平滑处理；中值滤波则是因为在每个滑动窗口区域内只取像素中值，其他像素舍弃，导致处理后的图像丢失了部分细节所致。

② 在中值滤波处理过程中，由于对原图像进行了 padding 处理，导致在图像四个边角处像素值为 0 的点过多，在取中值时返回值为 0，因此处理后的图像在四个角的区域部分为黑色。

③ 在双边滤波处理过程中，值域核中各个参数值的计算公式如下

$$r(i, j, k, l) = \exp\left(-\frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2}\right)$$

其中 σ_r^2 作为分母需要对其进行分母不为 0 处理。

通过分析可知， σ_r^2 为卷积区域（假设 5*5）内各个像素值的方差，当该区域方差为 0 时，

说明在该卷积区域内每个像素的像素值相等。针对这种情况有两种处理方案：第一种处理方案是直接将该区域内中间的像素值作为返回值输出；第二种处理方案是直接将最后的权重矩阵（weight）的每个参数设为 1，即代表邻域内每个像素点对最终输出值影响程度一样。

为了减少计算量，实验过程中采用了方案一的处理办法。