数字图像处理

第六次作业

翟宏佳

电信钱 61

2160405066

摘要：

本文首先实现图像的噪声退化和运动模糊，并选择不同参数以便于讨论各参数的具体作用。在图像复原方面，本文着重讨论维纳滤波和约束最小二乘方滤波，然后利用这两种方法对退化图像进行复原，对比两种方法的处理效果以得出各自的优缺点。

一、产生高斯噪声并恢复图像

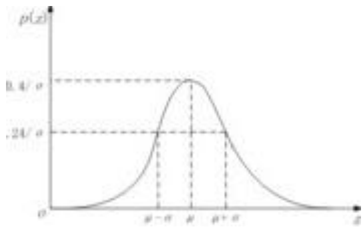
1、问题分析

高斯噪声：

所谓高斯噪声是指它的概率密度函数服从高斯分布（即正态分布）的一类噪声。一个高斯随机变量 z 的 PDF 可表示为：

$$P(z) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(z-u)^2}{2\sigma^2}\right]$$

其中 z 代表灰度，u 是 z 的均值，是 z 的标准差。高斯噪声的灰度值多集中在均值附近。



算术均值滤波器：

令 Sxy 表示中心在点(x,y)处，大小为 m×n 的矩形子图像窗口的一组坐标。算术均值滤波器在 Sxy 定义的区域中计算被污染的图像 g(x,y)的平均值。在点(x,y)处复原图像的值：

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

这个操作可以使用大小为 m×n 的一个空间滤波器来实现，其所有系数均为其值的 1/mn。均值滤波器平滑一幅图像中的局部变化，虽然模糊了结果，但降低了噪声。

中值滤波器：

统计排序滤波器是空间域滤波器，空间滤波器的响应是基于由该滤波器包围的图像区域中的像素值的排序。中值滤波器使用一个像素邻域中灰度级的中值来替代该像素值，即：

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

谐波均值滤波器：

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

逆谐波均值滤波器：

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Matlab 函数介绍：

采用函数 imnoise 来使用噪声污染一幅图像，该函数的基本语法为：g=imnoise(f,type,parameters)

常用噪声类型

'gaussian':Gaussian white noise with constant mean and variance

'localvar':Zero-mean Gaussian white noise with an intensity-dependentvariance

'poisson':Poisson noise

'salt & pepper':On and off pixels

'speckle':Multiplicative noise'gaussian'

语法

g = imnoise(I,'gaussian',m,v)

g = imnoise(I,'salt & pepper',d)

格式说明

f 为是输入图像。函数 imnoise 在给图像添加噪声之前，将它转换为范围[0,1]内的 double 类图像。指定噪声参数时必须考虑到这一点。

g=imnoise(f,'gaussian',m,var)将均值 m,方差为 var 的高斯噪声加到图像 f 上，默认值是均值 m 为 0，方差 var 为 0.01 的噪声。

g=imnoise(f,'localvar',V)将均值为 0，局部方差为 V 的高斯噪声添加到图像 f 上，其中 V 是与 f 大小相同的一个数组，它包含了每一个点的理想方差值。

g=imnoise(f,'localvar',image_intensity,var)将均值为 0 的高斯噪声添加到图像 f 中，其中噪声的局部方差 var 是图像 f 的亮度值的函数。参量 image_intensity 和 var 是大小相同的向量，plot(image_intensity,var)绘制出噪声方差和图像亮度的函数关系。向量 image_intensity 必须包含范围在[0,1]内的归一化亮度值。

g=imnoise(f,'salt&pepper',d)用椒盐噪声污染图像 f，其中 d 是噪声密度(即包括噪声值的图像区域的百分比)。因此，大约有 d*numel(f)个像素受到影响。默认的噪声密度为 0.05。

g=imnoise(f,'speckle',var)用方程 g=f+n*f)将乘性噪声添加到图像 f 上，其中 n 是均值为 0，方差为 var 的均匀分布的随机噪声，var 的默认值是 0.04。

g=imnoise(f,'poisson')从数据中生成泊松噪声，而不是将人工的噪声添加到数据中，为了遵守泊松统计，unit8 和 unit16 类图像的亮度必须和光子的数量相符合。当每个像素的光子数量大于 65535 时，就要使用双精度图像。亮度值在 0 到 1 之间变化，并且对应于光子的数量除以 10e12。

2、 实验结果

1>添加高斯噪声

lena.bmp

lena加入gaussian噪声后(u=0, s^2=0.01).bmp

lena加入gaussian噪声后(u=0, s^2=0.1).bmp

lena加入 gaussian噪声后 (u=0, s^2=0.05).bmp　lena加入 gaussian噪声后 (u=0.1, s^2=0.01).bmp　lena加入 gaussian噪声后 (u=0.5, s^2=0.01).bmp

2>图像恢复（选取被均值为 0，方差为 0.01 的高斯噪声污染的图像为例）

a)算术均值滤波器

b)中值滤波器

c)谐波均值滤波器

d)逆谐波均值滤波器

3、 结果分析

①首先通过 imnoise 函数分别产生了被不同均值和方差的高斯噪声污染的图像。当高斯噪声均值不变为 0 时，随着方差增加，图像噪声越严重；当高斯噪声方差不变时，均值会影响到整个图像的灰度值，使整个图像变亮。与理论上均值和方差对图像的影响一致。
②分别使用算术均值滤波器和中值滤波器对加噪图像进行恢复。两种方法在一定程度上都可以降低噪声。算术均值滤波器降低噪声的同时也模糊了图像。
③谐波均值滤波器和逆谐波均值滤波器都有一定的恢复效果，并且谐波均值滤波器对于高斯噪声的恢复效果更好一些，但是不如中值滤波器。

二、产生椒盐噪声并恢复图像

1、问题分析

椒盐噪声：

椒盐噪声（salt & pepper noise）是数字图像的一个常见噪声，所谓椒盐。椒就是黑，盐就是白，椒盐噪声就是在图像上随机出现黑色白色的像素。椒盐噪声是一种由于

信号脉冲强度引起的噪声，产生该噪声的算法也比较简单。

给一副数字图像加上椒盐噪声的过程例如以下：

（1）指定信噪比 SNR （其取值范围在[0, 1]之间）

（2）计算总像素数目 SP。 得到要加噪的像素数目 NP = SP * (1-SNR)

（3）随机获取要加噪的每一个像素位置 P（i, j）

（4）指定像素值为 255 或者 0。

（5）反复 3,4 两个步骤完毕全部像素的 NP 个像素

（6）输出加噪以后的图像

有关滤波器在上一问题中介绍过，不再重复。

2、实验结果

1>添加椒盐噪声

2>图像恢复

a)算数均值滤波器

b)中值滤波器

c)谐波均值滤波器

d)逆谐波均值滤波器

图 1 加入椒盐噪声


图 2 Q=-0.5


图 3 Q=-1.5


图 4 Q=0.5


图 5 Q=1.5

3、结果分析

①首先通过 imnoise 函数产生了被椒和盐噪声密度均为 0.1 的椒盐噪声污染的图像，再分别使用算术均值滤波器和中值滤波器对加噪图像进行恢复。两种方法在一定程度上都可以降低噪声。但可以发现，中值滤波效果较好，而算数均值滤波器在一定程度上模糊了图像。

②再分别使用谐波均值滤波器和逆谐波均值滤波恢复图像，可以发现谐波均值滤波器不能去除椒盐噪声，而逆谐波均值滤波器，可以通过 Q 值的变化，来获得一定的效果。当 Q 为正，这个滤波器可以去除胡椒噪声，当 Q 为负，这个滤波器可以去除盐粒噪声。

三、推导维纳滤波器并实现以下要求

(a) 实现模糊滤波器如方程 Eq. (5.6-11).

(b) 模糊 lena 图像：45 度方向，T=1；

(c) 再模糊的 lena 图像中增加高斯噪声，均值= 0 ，方差=10 pixels 以产生模糊图像；

(d) 分别利用方程 Eq. (5.8-6)和(5.9-4)，恢复图像；并分析算法的优缺点

1、问题分析

1）维纳滤波器的推导：

图像的退化模型为：

$$x(\mathrm{n}_1,\mathrm{n}_2)=b(\mathrm{n}_1,\mathrm{n}_2)*s(\mathrm{n}_1,\mathrm{n}_2)+\mathrm{w}(\mathrm{n}_1,\mathrm{n}_2) \quad (1)$$

其中，s(n1,n2)为原始图像，b(n1,n2)为退化函数，w(n1,n2)为噪声函数，x(n1,n2)为退化的图像。并假设 s 与 w 不相关，w 为 0 均值的平稳随机过程。

图像的复原模型为：

$$\hat{s}(\mathrm{n}_1,\mathrm{n}_2)=\mathrm{h}(\mathrm{n}_1,\mathrm{n}_2)*x(\mathrm{n}_1,\mathrm{n}_2)=\sum_{l_1}\sum_{l_2}h(\mathrm{l}_1,\mathrm{l}_2)\times\mathrm{x}(\mathrm{n}_1-\mathrm{l}_1,\mathrm{n}_2-\mathrm{l}_2) \quad (2)$$

其中，$\hat{s}(\mathrm{n}_1,\mathrm{n}_2)$为恢复的图像，$\mathrm{h}(\mathrm{n}_1,\mathrm{n}_2)$为恢复滤波器。

误差度量为：

$$e^2=E\{(s(\mathrm{n}_1,\mathrm{n}_2)-\hat{s}(\mathrm{n}_1,\mathrm{n}_2))^2\} \quad (3)$$

基于正交性原理，若要求误差最小，则必有下式成立：

$$E\{\mathrm{e}(\mathrm{n}_1,\mathrm{n}_2)\times x^*(\mathrm{m}_1,\mathrm{m}_2)\}=0 \quad (4)$$

将（3）式带入（4）式有：

$$E\{\mathrm{s}(\mathrm{n}_1,\mathrm{n}_2)\times x^*(\mathrm{m}_1,\mathrm{m}_2)\}=E\{\hat{\mathrm{s}}(\mathrm{n}_1,\mathrm{n}_2)\times x^*(\mathrm{m}_1,\mathrm{m}_2)\} \quad (5)$$

即

$$\begin{aligned}
&R_{sx}(\mathrm{n}_1-\mathrm{m}_1,\mathrm{n}_2-\mathrm{m}_2)=E\{\sum_{l_1}\sum_{l_2}h(\mathrm{l}_1,\mathrm{l}_2)\times\mathrm{x}(\mathrm{n}_1-\mathrm{l}_1,\mathrm{n}_2-\mathrm{l}_2)\times x^*(\mathrm{m}_1,\mathrm{m}_2)\}\\
&=\sum_{l_1}\sum_{l_2}h(\mathrm{l}_1,\mathrm{l}_2)\times E\{\mathrm{x}(\mathrm{n}_1-\mathrm{l}_1,\mathrm{n}_2-\mathrm{l}_2)\times x^*(\mathrm{m}_1,\mathrm{m}_2)\}\\
&=\sum_{l_1}\sum_{l_2}h(\mathrm{l}_1,\mathrm{l}_2)\times R_{xx}(\mathrm{n}_1-\mathrm{l}_1-\mathrm{m}_1,\mathrm{n}_2-\mathrm{l}_2-\mathrm{m}_2)\\
&=h(\mathrm{n}_1-\mathrm{m}_1,\mathrm{n}_2-\mathrm{m}_2)*R_{xx}(\mathrm{n}_1-\mathrm{m}_1,\mathrm{n}_2-\mathrm{m}_2)
\end{aligned}$$

（6）

换元得：

$$R_{sx}(\mathrm{n}_1,\mathrm{n}_2)=h(\mathrm{n}_1,\mathrm{n}_2)*R_{xx}(\mathrm{n}_1,\mathrm{n}_2) \quad (7)$$

等式两端同时取傅里叶变换得：

$$P_{sx}(\mathrm{w}_1,\mathrm{w}_2)=H(\mathrm{w}_1,\mathrm{w}_2)\times P_x(\mathrm{w}_1,\mathrm{w}_2) \quad (8)$$

即

$$H(\mathrm{w}_1,\mathrm{w}_2)=\frac{P_{sx}(\mathrm{w}_1,\mathrm{w}_2)}{P_x(\mathrm{w}_1,\mathrm{w}_2)} \quad (9)$$

公式（8）中

$$
\begin{aligned}
&R_{sx}(n_1,n_2)=E\{s(n_1+k_1,n_2+k_2)\times x^*(k_1,k_2)\}\\
&=E\{s(n_1+k_1,n_2+k_2)\times(\sum_{l_1}\sum_{l_2}b(l_1,l_2)\times s(k_1-l_1,k_2-l_2)+w(k_1,k_2))^*\}\\
&=\sum_{l_1}\sum_{l_2}b^*(l_1,l_2)\times E\{s(n_1+k_1,n_2+k_2)\times s^*(k_1-l_1,k_2-l_2)+s(n_1+k_1,n_2+k_2)\times w^*(k_1,k_2)\}\\
&=\sum_{l_1}\sum_{l_2}b^*(l_1,l_2)\times R_s(n_1+l_1,n_2+l_2)\\
&=b^*(-n_1,-n_2)*R_s(n_1,n_2)
\end{aligned}
$$

（10）

公式（10）两端同时取傅里叶变换得：

$$P_{sx}(w_1,w_2)=B^*(w_1,w_2)\times P_s(w_1,w_2) \quad （11）$$

公式（8）中

$$
\begin{aligned}
&R_x(n_1,n_2)=E\{x(n_1+k_1,n_2+k_2)\times x^*(k_1,k_2)\}\\
&=E\{(\sum_{l_1}\sum_{l_2}b(l_1,l_2)\times s(n_1+k_1-l_1,n_2+k_2-l_2)+w(n_1+k_1,n_2+k_2))\times(\sum_{m_1}\sum_{m_2}b(m_1,m_2)\times s(k_1-m_1,k_2-m_2)+w(k_1,k_2))^*\}\\
&=\sum_{l_1}\sum_{l_2}\sum_{m_1}\sum_{m_2}b(l_1,l_2)\times b^*(m_1,m_2)\times E\{s(n_1+k_1-l_1,n_2+k_2-l_2)\times s^*(k_1-m_1,k_2-m_2)+R_w(n_1,n_2)\\
&=\sum_{m_1}\sum_{m_2}b^*(m_1,m_2)\sum_{l_1}\sum_{l_2}b(l_1,l_2)R_s(n_1+m_1-l_1,n_2+m_2-l_2)+R_w(n_1,n_2)\\
&=\sum_{m_1}\sum_{m_2}b^*(m_1,m_2)\times(b(n_1+m_1,n_2+m_2)*R_s(n_1+m_1,n_2+m_2))+R_w(n_1,n_2)\\
&=b^*(-n_1,-n_2)*b(n_1,n_2)*R_s(n_1,n_2)+R_w(n_1,n_2)
\end{aligned}
$$

（12）

公式（12）两端同时取傅里叶变换：

$$P_x(w_1,w_2)=\left|B(w_1,w_2)\right|^2\times P_s(w_1,w_2)+P_w(w_1,w_2) \quad （13）$$

将（11）式和（13）式带入（8）式得

$$H(w_1,w_2)=\frac{B^*(w_1,w_2)\times P_s(w_1,w_2)}{\left|B(w_1,w_2)\right|^2\times P_s(w_1,w_2)+P_w(w_1,w_2)} \quad （14）$$

将符号化成与书中一致的表示

$$
\begin{aligned}
w(u,v)&=\frac{H^*(u,v)\times\left|F(u,v)\right|^2}{\left|H(u,v)\right|^2\times\left|N(u,v)\right|^2+\left|N(u,v)\right|^2}\\
&=\frac{H^*(u,v)}{\left|H(u,v)\right|^2+\dfrac{\left|N(u,v)\right|^2}{\left|F(u,v)\right|^2}}
\end{aligned}
$$

（15）

故表达式由下式给出

$$
\begin{aligned}
\hat{\mathrm{F}}(\mathrm{u,v}) &= [\frac{H^*(\mathrm{u,v}) \times S_f(\mathrm{u,v})}{|H(\mathrm{u,v})|^2 \times S_f(\mathrm{u,v}) + S_\eta(\mathrm{u,v})}]\mathrm{G}(\mathrm{u,v}) \\
&= [\frac{H^*(\mathrm{u,v})}{|H(\mathrm{u,v})|^2 + S_\eta(\mathrm{u,v}) / S_f(\mathrm{u,v})}]\mathrm{G}(\mathrm{u,v}) \\
&= [\frac{1}{H(\mathrm{u,v})} \frac{|H(\mathrm{u,v})|^2}{|H(\mathrm{u,v})|^2 + S_\eta(\mathrm{u,v}) / S_f(\mathrm{u,v})}]\mathrm{G}(\mathrm{u,v})
\end{aligned}
$$

（16）

2）约束最小二乘方滤波

对于约束最小二乘方滤波，期望是找一个最小准则函数 C，定义如下：

$$C = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\nabla^2 \mathrm{f(x,y)}]^2 \tag{17}$$

其约束为

$$\left\| g - H\hat{f} \right\|^2 = \|\eta\|^2 \tag{18}$$

其中，$\|w\|^2 \triangleq w^T w$ 是欧几里得向量范数，$\hat{f}$ 是未退化图像的估计。

这个最佳问题在频率域中的解决由下面的表达式给出：

$$\hat{F}(\mathrm{u,v}) = [\frac{H^*(\mathrm{u,v})}{|H(\mathrm{u,v})|^2 + \gamma |P(\mathrm{u,v})|^2}]\mathrm{G}(\mathrm{u,v}) \tag{19}$$

其中，γ 是一个参数，必须对它进行调整以满足式（18）的条件，P(u,v)是函数

$$p(\mathrm{x,y}) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{20}$$

的傅里叶变换。

2、实验结果

a）实现模糊滤波器如方程 Eq. (5.6-11).

模糊滤波器的频域表达式为：

$$H(u,v) = \frac{T}{\pi(ua+vb)}\sin[\pi(ua+vb)]e^{-j\pi(ua+vb)}$$

故实现该滤波器，只需先将输入图像进行傅里叶变换并移至图像中心，之后将图像的傅里叶变换和模糊滤波器的傅里叶变换进行阵列相乘，将得到的结果经过傅里叶反变换返回到空间域即可实现该滤波器。

b）模糊 lena 图像：45 度方向，T=1；(a=0.1，b=0.1；T=1)

lena.bmp 原始图像　　　　　　　lena 运动模糊的结果（a=0.1，b=0.1；T=1）

lena.bmp 原始图像　　lena 运动模糊的结果（调用 MATLAB 函数）

（c）在模糊的 lena 图像中增加高斯噪声，均值= 0 ，方差=10 pixels 以产生模糊图像；

lena 运动模糊的结果(a=0.1, b=0.1 ;T=1)　　添加高斯噪声的结果(均值为 0, 方差为 0.01)

lena 运动模糊的结果（MATLAB 版）　　添加高斯噪声的结果（MATLAB 版）


（d）分别利用方程 Eq. (5.8-6)和(5.9-4)，恢复图像。

维纳滤波的结果：

lena 运动模糊+高斯噪声.bmp

维纳滤波结果（K=0.06）

维纳滤波结果（K=0.01）

维纳滤波结果（K=0.5）

lena 运动模糊+高斯噪声（MATLAB 版） 维纳滤波结果（MATLAB 版）

约束最小二乘方滤波的结果：

lena 运动模糊+高斯噪声（MATLAB 版） 约束最小二乘滤波结果（MATLAB 版）

3、结果分析

①首先分别通过自己编写的模糊函数和 MATLAB 中提供的 imfilter 和 fspecial 函数配合使用对图像 lena 进行了模糊滤波。发现套用书上的公式图像是斜向下 45 度运动模糊，而 MATLAB 中函数模糊的结果是斜向上 45 度运动模糊，不过这不是重点可以接受，模糊的基本效果还是一致的。之后调用 imnoise 函数对两幅图像加入高斯噪声，得到第二问的结果。

②之后分别使用自己编写的函数和 MATLAB 中提供的 deconvwnr 函数进行维纳滤波。调用 MATLAB 中函数滤波后的图像得到了一定的改善，运动模糊的影响基本被消除，但噪声的影响仍然较大，导致图像质量下降；对于自己编写的维纳滤波函数，难点在于寻找令信噪比最大的 K 值，报告中显示了部分 K 值对应的滤波结果，其中 K=0.06，为信噪比最大时的滤波结果，从结果看，视觉上的效果并不是很理想，要想达到更好的效果可能需要寻找更加合适的 K 值。

③最后采用 MATLAAB 提供的 deconvreg 函数进行约束最小二乘方滤波。从滤波后的结果看，约束最小二乘方滤波得到了比维纳滤波更好的结果，尤其是对噪声的滤除。