Обзор статей про восстановление размытых изображений

Процесс искажения изображений можно представить в виде формулы
G(u,v)=H(u,v)*F(u,v)+N(u,v)
Где:
G(u,v)- искаженное изображение
H(u,v)-искажающая функция
F(u,v) -искажаемое изображение
N(u,v)-адаптивный шум
*-операция свертки

Некоторые модели шумов

1) Гауссов шум

Функция плотности распределения гуассовой случайной величины z задается выражением

$$p(z)=\frac{1}{\sigma\sqrt{2\pi}}e^{-(z-\mu)^2/2\sigma^2}$$

Где: z – значение яркости
μ - среднее значение случайно величины z
σ -ее среднеквадратичное отклонение

2)Шум Релея

функция плотности распределения вероятностей шума Релея задается выражением

$$p(z)=\begin{cases}\frac{2}{b}(z-a)e^{-(z-a)^2/b}\; при\; z\geq a\\ 0\; при\; z<a\end{cases}$$

где среднее и дисперсия имеют вид

$$\mu=a+\sqrt{(\pi b/4)}$$

$$\sigma^2=\frac{b(4-\pi)}{4}$$

3)Шум Эрланга

$$p(z)=\begin{cases}\frac{a^b z^{(b-1)}}{(b-1)!}\; при\; z\geq 0\\ 0\; при\; z<0\end{cases}$$

где a>0, b -положительное целое число

$$\mu=\frac{b}{a}$$

$$\sigma^2=\frac{b}{a^2}$$

4)Экспоненциальный шум

$$p(z)=\begin{cases}a*e^{-az}\; при\; z\geq 0\\ 0\; при\; z<0\end{cases}$$

где a>0
и среднее и дисперсия имеют вид

$$\mu=\frac{1}{a}$$

$$\sigma^2=\frac{1}{a^2}$$

По сути, это распределение Эрланга с b=1

5)Равномерный шум

$$p(z)=\begin{cases} \dfrac{1}{b-a}\,при\,a\leq z\leq b \\ 0,\,в\,остальных\,случаях \end{cases}$$

среднее значение и дисперсия равны

$$\mu=\frac{a+b}{2}$$

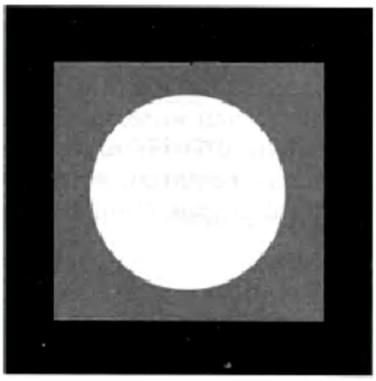
$$\sigma^2=\frac{(b-a)^2}{12}$$

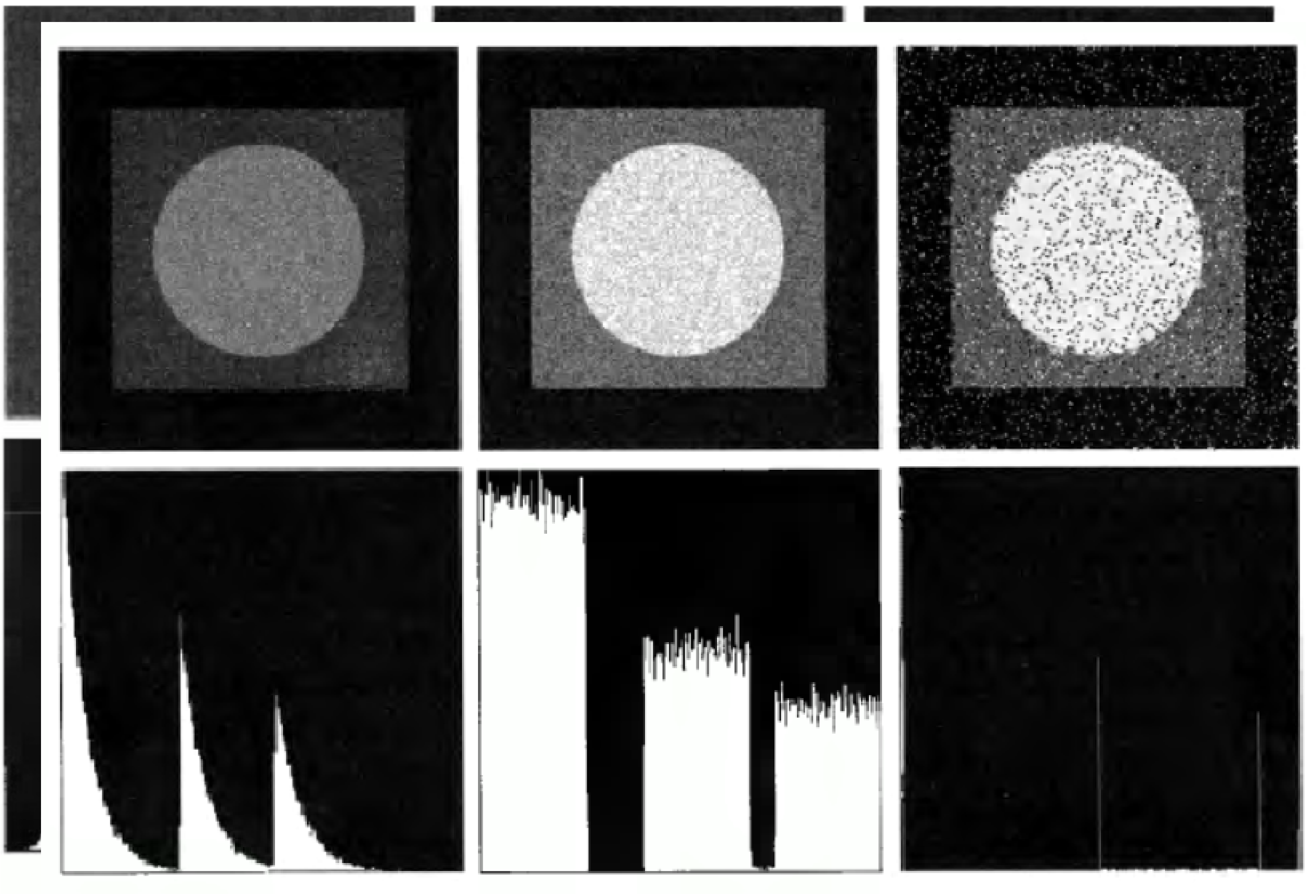
6)Импульсный шум

$$p(z)=\begin{cases} P_a,\,z=a \\ P_b,\,z=b \\ 0,\,иначе \end{cases}$$

Эти распределения представляют собой средства для моделирования искажений

Исходное изображение:





Экспоненциальный шум | Равномерный шум | Импульсный шум

Если изображение искажалось только шумами, то его вид следующий

$$g(x)=f(x,y)+n(x,y)$$

$$G(u,v)=F(u,v)+N(u,v)$$

Для подавления шумов существуют разные фильтры, основанные на порядковых статистиках, т. е. В этих фильтрах считаются среднее, дисперсия, медиана и еще необходимые данные , и в зависимости от них изменяются значения согласно тому или иному фильтру.

Адаптивные фильтры

Существуют так же адаптивные фильтры, они так называются, т. к. их поведение меняется в зависимости от статистических свойств изображения внутри прямоугольной области $m \times n$ в окрестности S_{xy} .

Рассмотрим следующий фильтр:

$$\hat{f}=g(x,y)-\frac{\sigma_\eta^2}{\sigma_L^2}[g(x,y)-m_L]$$

где

$\hat{f}$ - восстановленное изображение

g(x,y) – искаженное изображение

σ_η^2 - дисперсия по всему изображению

σ_L^2 - дисперсия по окрестности S_{xy}

m_L - среднее по окрестности S_{xy}

если дисперсия по окрестности больше дисперсии по всему изображению, то это отношение равно 1

Попытка реализовать этот фильтр на Python

```
def addapt_loc_filter():
    im=plb.imread("shema.jpg")
    bwi=make_black_white_im(im)
    n=7
    h=len(im[:])
    w=len(im[0,:])
    bw=np.zeros((h,w,3))
    bw[:,:,0]=bwi
    bw[:,:,1]=bwi
    bw[:,:,2]=bwi
    plb.imsave("shema_wb.jpg",bw)
    d_gl=disp(bwi)
    print h,w
    for i in range(n/2,h-n/2):
        for j in range(n/2,w-n/2):
            m=mean(bwi[i-n//2:i+n//2+1,j-n//2:j+n//2+1])
            d=disp(bwi[i-n//2:i+n//2+1,j-n//2:j+n//2+1])
            k=0
            if (d< d_gl):
                k = 1
            else:
                k = float(d_gl)/d
            bw[i,j,0]=bwi[i,j]-k*(bwi[i,j]-m)
            bw[i, j, 1] = bwi[i, j] - k * (bwi[i, j] - m)
            bw[i, j, 2] = bwi[i, j] - k * (bwi[i, j] - m)
    plb.imsave("addapt_loc_filter/res6.jpg",bw)
```

Вычисление среднего

```
def mean(pix):
    a=np.sum(pix)
```

```
    a/=float(pix.size)
    return a
```

Вычисление дисперсии

def disp(pix):

```
    m=mean(pix)
    res=np.sum((m-pix)**2)
    res=res*(1./pix.size)
    return res
```

Фильтрация методом минимизация среднеквадратического отклонения (винеровская фильтрация)

Фильтр Винера является методом, соединяющий в себе учет свойств искажающей функции и статистических свойств шума в процессе восстановления. Метод основан на рассмотрении изображений и шума как случайных процессов, и задача ставится следующим образом: найти такую оценку $\hat{f}$ для неискаженного изображения f чтобы среднеквадратичное отклонение этих величин было минимальным. Стандартное отклонение задается следующей формулой:

$$e^2 = E\left\{(f-\hat{f})^2\right\}$$

Предполагается, что выполнены условия:
1) шум и неискаженное изображение некоррелированы между собой
2) либо шум, либо неискаженное изображение имеют нулевое среднее значение
3) оценка линейно зависит от искаженного изображения. При выполнении этих условий минимум среднеквадратичного отклонения достигается на функции, которая задается в частотной области выражением

$$\hat{F}(u,v)=\left(\frac{H'(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2+S_\eta(u,v)}\right)G(u,v)=\left(\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+S_\eta(u,v)/S_f(u,v)}\right)G(u,v)$$

Где:
$H(u,v)$ - искажающая функция (ее частотное представление)
$H'(u,v)$ - комплексное сопряженное искажающей функции
$|H(u,v)|^2=H'(u,v)H(u,v)$
$S_\eta=|N(i,v)|^2$ - энергетический спектр шума
$S_f(u,v)=|F(u,v)|^2$ - энергетический спектр неискаженного изображения
$G(u,v)$ - фурье-преобразование искаженного изображения

Последнее равенство имеет место, из-за того, что произведение комплексного числа на комплексно-сопряженное равно квадрату модуля
Этот результат был получен Винером, и этот результат известен как оптимальная фильтрация по Винеру. Фильтр внутри скобок часто называют фильтром среднеквадратического отклонения или винеровским фильтром.
Когда спектр шума и неискаженного изображения неизвестно и не могут быть оценены, часто пользуются подходом, состоящий в предыдущего выражения следующим

$$\hat{F}(u,v)=\left(\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+K}\right)G(u,v)$$

где K- определенная константа.