

# CVPR 第一次作业

强基数学 002 吴天阳 2204210460

## 1.1 实验 1-1 理论推导

**定义 1.** 我们将  $w \in \mathbb{R}^{2k+1, 2k+1}$  称为大小等于  $(2k+1, 2k+1)$  的卷积核 (卷积核长宽一般为奇数且相等), 图像记为  $f \in \mathbb{R}^{M \times N}$  (这里可将图像  $f$  视为函数,  $f(m, n)$  表示在点  $(m, n)$  处的像素值), 则互相关  $S[f]$  在  $(m, n)$ ,  $(m \leq M, n \leq N)$  处的定义为

$$S[f](m, n) = w \otimes f(m, n) = \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} w_{ij} f(m+i-1, n+j-1).$$

卷积  $S[f]$  在  $(m, n)$ ,  $(m \leq M, n \leq N)$  处的定义为

$$S[f](m, n) = w * f(m, n) = \sum_{i=1}^{2k+1} w_{ij} \sum_{j=1}^{2k+1} f(m-i+1, n-j+1).$$

**注解 1.** 若  $f(x, y)$  的取值超出了图像的取值范围, 即  $x \notin [1, M]$  或  $y \notin [1, N]$ , 则取  $f(x, y) = 0$ , 也称为零填充操作 (可将  $f$  理解为原图像在  $\mathbb{R}^2$  上的零延拓).

于是在零延拓的意义下可以将上述定义写为

$$\begin{aligned} w \otimes f(m, n) &= \sum_{i, j \in \mathbb{Z}} w_{ij} f(m+i-1, n+j-1), \\ w * f(m, n) &= \sum_{i, j \in \mathbb{Z}} w_{ij} f(m-i+1, n-j+1). \end{aligned}$$

**命题 1 (线性性).** 设  $f, g$  为两幅不同的图像,  $a, b$  为系数,  $f' = af + bg$  则

$$w \otimes f' = a(w \otimes f) + b(w \otimes g).$$

设滤波核  $w' = aw + bv$ ,  $w, v$  为两个不同的滤波核,  $a, b$  为系数, 则

$$w' \otimes f = a(w \otimes f) + b(v \otimes f).$$

证明.

$$\begin{aligned} w \otimes f'(m, n) &= w \otimes (af + bg)(m, n) = \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} w_{ij} (af + bg)(m+i-1, n+j-1) \\ &\stackrel{\text{由函数之间的线性性}}{=} \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} w_{ij} (af(m+i-1, n+j-1) + bg(m+i-1, n+j-1)) \\ &= \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} aw_{ij} f(m+i-1, n+j-1) + bw_{ij} g(m+i-1, n+j-1) \\ &= a(w \otimes f)(m, n) + b(w \otimes g)(m, n). \end{aligned}$$

由命题 3 的对称性可知,  $w' \otimes f = f \otimes w' = a(f \otimes w) + b(f \otimes v) = a(w \otimes f) + b(v \otimes f)$ .  $\square$

**命题 2** (平移不变性). 设  $f$  平移后的的图像  $f'(m, n) = f(m - m_0, n - n_0)$ , 证明

$$(w \otimes f')(m, n) = (w \otimes f)(m - m_0, n - n_0).$$

证明.

$$\begin{aligned} (w \otimes f')(m, n) &= \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} w_{ij} f'(m + i - 1, n + j - 1) \\ &= \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} w_{ij} f(m - m_0 + i - 1, n - n_0 + j - 1) \\ &= (w \otimes f)(m - m_0, n - n_0). \end{aligned}$$

□

**命题 3** (交换律, 结合律). 交换律:  $w * f = f * w$ , 结合律:  $v * (w * f) = (v * w) * f$ .

证明. 交换律

$$\begin{aligned} w * f(m, n) &= \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} w_{ij} f(m - i + 1, n - j + 1) \\ &\stackrel{\substack{i'=m-i+1 \\ j'=n-j+1}}{=} \sum_{i'=m-2k}^m \sum_{j'=n-2k}^n w(m - i' + 1, n - j' + 1) f_{i'j'} \\ &= \sum_{i'=1}^M \sum_{j'=1}^N f_{i'j'} w(m - i' + 1, n - j' + 1) \\ &= f * w(m, n) \end{aligned} \tag{1}$$

其中 (1) 式是因为  $w(x, y)$  表示矩阵  $w$  在  $\mathbb{R}^2$  上的零延拓, 则  $\text{supp}(w) = \{(x, y) : x, y = 1, 2, \dots, 2k + 1\}$ ,  $\text{supp}(w)$  表示  $w$  取值不为零的点所成的集合的闭包. 于是使得  $m - i' + 1 \in [1, 2k + 1]$ , 则  $i' \in [m - 2k, m]$ , 所以 (1) 式成立.

结合律:

$$\begin{aligned} v * (w * f)(m, n) &= \sum_{i, j \in \mathbb{Z}} v_{ij} (w * f)(m - i + 1, n - j + 1) \\ &= \sum_{i, j \in \mathbb{Z}} v_{ij} \sum_{i', j' \in \mathbb{Z}} w_{i'j'} f(m - i - i' + 2, n - j - j' + 2) \\ &\stackrel{\substack{i'=s-i+1 \\ j'=t-j+1}}{=} \sum_{s, t \in \mathbb{Z}} \sum_{i, j \in \mathbb{Z}} v_{ij} w(s - i + 1, t - j + 1) f(m - s + 1, n - t + 1) \\ &= \sum_{s, t \in \mathbb{Z}} (v * w(s, t)) f(m - s + 1, n - t + 1) = (v * w) * f. \end{aligned}$$

□

**命题 4.** 设定图像大小为  $M \times N$ , 滤波核大小为  $(2k + 1) \times (2k + 1)$ , 则 2D 卷积操作的时间复杂度为  $\mathcal{O}(MN(2k + 1)^2)$ .

证明. 计算每个卷积核时间复杂度为  $\mathcal{O}((2k+1)^2)$ , 若使用等宽卷积, 则需要计算  $MN$  个卷积核, 故总复杂度为  $\mathcal{O}(MN(2k+1)^2)$ .  $\square$

**命题 5 (2D 高斯核可分离性).** 令标准差为  $\sigma$  的一维高斯函数  $G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ , 二维高斯函数  $G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = G_\sigma(x)G_\sigma(y)$ , 如果将  $G_\sigma(x)$  转化为  $(2k+1) \times 1$  的滤波器  $u$ ,  $G_\sigma(y)$  转化为  $1 \times (2k+1)$  的滤波器  $v$ , 则二维高斯核为  $w = u * v$ .

证明. 滤波器是以高斯函数的均值作为中心的插值结果, 可将一维高斯核定义为  $u_i = G_\sigma(i - k - 1)$ , 二维高斯核定义为  $w_{ij} = G_\sigma(i - k - 1, j - k - 1)$ , 则

$$\begin{aligned} u * v(i, j) &= \sum_{\substack{1 \leq s \leq 2k+1 \\ i-s+1=1}} \sum_{\substack{t=1 \\ 1 \leq j-t+1 \leq 2k+1}} u_{st} v_{i-s+1, j-t+1} = u_{i1} v_{1j} \\ &= G_\sigma(i - k - 1) G_\sigma(j - k - 1) = w_{ij}. \end{aligned}$$

$\square$

**命题 6.** 两个 2D 高斯函数的卷积仍为一个 2D 高斯函数

$$(G_{\sigma_1} * G_{\sigma_2})(m, n) = G_\sigma(m, n)$$

则  $\sigma^2 = \sigma_1^2 + \sigma_2^2$ .

证明.

$$\begin{aligned} (G_{\sigma_1} * G_{\sigma_2})(m, n) &= \frac{1}{(2\pi)^2 \sigma_1^2 \sigma_2^2} \int_{\mathbb{R}} dx \int_{\mathbb{R}} dy e^{-\frac{x^2+y^2}{2\sigma_1^2}} e^{-\frac{(m-x)^2+(n-y)^2}{2\sigma_2^2}} \\ &= \frac{1}{(2\pi)^2 \sigma_1^2 \sigma_2^2} \int_{\mathbb{R}} dx \int_{\mathbb{R}} dy e^{-\frac{x^2+y^2}{2\sigma_1^2}} e^{-\frac{x^2-2mx+m^2+y^2-2ny+n^2}{2\sigma_2^2}} \\ &= \frac{1}{(2\pi)^2 \sigma_1^2 \sigma_2^2} \int_{\mathbb{R}} dx \int_{\mathbb{R}} dy e^{-\frac{(\sigma_1^2+\sigma_2^2)x^2-2m\sigma_1^2x+(\sigma_1^2+\sigma_2^2)y^2-2n\sigma_1^2y}{2\sigma_1^2\sigma_2^2}} e^{-\frac{\sigma_1^2(m^2+n^2)}{2\sigma_1^2\sigma_2^2}} \\ &= \frac{1}{(2\pi)^2 \sigma_1^2 \sigma_2^2} \int_{\mathbb{R}} dx \int_{\mathbb{R}} dy e^{-\frac{\left(x^2 - \frac{m\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)^2 + \left(y^2 - \frac{n\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}} e^{-\frac{\sigma_1^2(m^2+n^2)\left(1 - \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)}{2\sigma_1^2\sigma_2^2}} \\ &= \frac{1}{2\pi(\sigma_1^2 + \sigma_2^2)} e^{-\frac{m^2+n^2}{\sigma_1^2+\sigma_2^2}} \int_{\mathbb{R}} dx \int_{\mathbb{R}} dy \frac{1}{2\pi \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}} e^{-\frac{\left(x^2 - \frac{m\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)^2 + \left(y^2 - \frac{n\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}} \\ &\stackrel{\text{二维高斯密度函数}}{\text{在}\mathbb{R}^2\text{上积分为1}} \frac{1}{2\pi(\sigma_1^2 + \sigma_2^2)} e^{-\frac{m^2+n^2}{\sigma_1^2+\sigma_2^2}} = G_\sigma(m, n) \end{aligned}$$

$\square$

**命题 7 (2D 空间卷积定理).** 设图像大小为  $M \times N$ ,  $h = f * g$ , 且  $F = \mathcal{F}(f)$ ,  $G = \mathcal{F}(g)$ ,  $H = \mathcal{F}(h)$ , 则  $H(k, l) = F(k, l)G(k, l)$ .

证明.

$$\begin{aligned}
 H(k, l) &= \sum_{m, n} f * g(m, n) e^{-2\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)} \\
 &= \sum_{m, n} \sum_{s, t} f(s, t) g(m - s, n - t) e^{-2\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)} \\
 &= \sum_{m, n} \sum_{s, t} f(s, t) e^{-2\pi i \left( \frac{ks}{M} + \frac{lt}{N} \right)} g(m - s, n - t) e^{-2\pi i \left( \frac{k(m-s)}{M} + \frac{l(n-t)}{N} \right)} \\
 &\stackrel{\substack{m-s=u \\ n-t=v}}{=} \sum_{s, t} f(s, t) e^{-2\pi i \left( \frac{ks}{M} + \frac{lt}{N} \right)} \sum_{u, v} g(u, v) e^{-2\pi i \left( \frac{ku}{M} + \frac{lv}{N} \right)} \\
 &= F(k, l) G(k, l)
 \end{aligned}$$

□

**命题 8 (2D 频域卷积定理).** 设图像大小为  $M \times N$ ,  $H = F * G$ , 且  $F = \mathcal{F}(f), G = \mathcal{F}(g), H = \mathcal{F}(h)$ , 则  $h(m, n) = af(m, n)g(m, n)$ .

证明.

$$\begin{aligned}
 h(m, n) &= \frac{1}{MN} \sum_{k, l} F * G(k, l) e^{2\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)} \\
 &= \frac{1}{MN} \sum_{k, l} \sum_{s, t} F(s, t) G(k - s, l - t) e^{2\pi i \left( \frac{km}{M} + \frac{ln}{N} \right)} \\
 &= \frac{1}{MN} \sum_{k, l} \sum_{s, t} F(s, t) e^{2\pi i \left( \frac{sm}{M} + \frac{tn}{N} \right)} G(k - s, l - t) e^{2\pi i \left( \frac{(k-s)m}{M} + \frac{(l-t)n}{N} \right)} \\
 &\stackrel{\substack{u=k-s \\ v=l-t}}{=} MN \frac{1}{MN} \sum_{s, t} F(s, t) e^{2\pi i \left( \frac{sm}{M} + \frac{tn}{N} \right)} \sum_{u, v} G(u, v) e^{2\pi i \left( \frac{um}{M} + \frac{vn}{N} \right)} \\
 &= MN \cdot f(m, n) g(m, n)
 \end{aligned}$$

□

## 1.2 实验目的

1. 理解线性滤波器性质, 了解线性滤波器核非线性滤波器之间的区别.
2. 编程实现高斯滤波, 实现并验证高斯滤波的特性, 设计锐化滤波器.
3. 编程实现双边滤波.
4. 掌握图像的 Fourier 变换, 编程实现图像在频域的高斯滤波试验.

## 1.3 实验原理

### 1.3.1 高斯滤波器

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

设和函数大小为  $2k + 1 \times 2k + 1$ , 对高斯函数在  $(i, j)$  ( $-k \leq i, j \leq k$ ) 点处进行采样, 再进行归一化处理.

**高斯核方差和滤波器大小关系** 利用高斯函数的性质，在距离中心  $\sigma$  中取到的体积占据总体积的 68%，在  $2\sigma$  中占比为 95%，在  $3\sigma$  中占比为 99.7%，如下图所示：

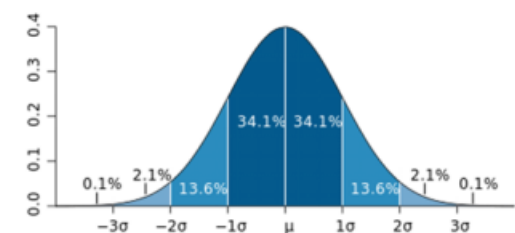


图 1:  $3\sigma$  原则

这里取  $2\sigma$  为高斯函数的边界，设滤波器大小为  $2k + 1 \times 2k + 1$ ，则  $\sigma = \frac{k}{2}$ 。以下为  $k = 1$ ，大小为  $3 \times 3$  的高斯滤波器

```
[0.01134374 0.08381951 0.01134374]
[0.08381951 0.61934703 0.08381951]
[0.01134374 0.08381951 0.01134374]
```

图 2:  $3 \times 3$  高斯核

### 1.3.2 设计边界处理方法

因为卷积使得图像的长宽减少  $2k$  ( $2k + 1$  为卷积核的大小)，所以为了保持前后图像大小一致，需要进行对图像进行填充操作，简单的填充方案有：零填充，镜像填充，边界填充，环绕填充。图3显示了零填充效果图，图4给出了在零填充后进行不同标准差的高斯核处理效果。（下文全部卷积操作均为做零填充的等大小卷积）

### 1.3.3 高斯核与高斯核的卷积

取用两个高斯核卷积得到的高斯核，与直接生成的二维高斯核，取出两个图像  $5 \times 5$  大小的中心部分做差取绝对值，可以得到两者差值非常小，可近似相等，说明两个标准差为  $\sigma_1, \sigma_2$  的卷积核，通过卷积可得到  $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$  的卷积核。实验效果如图5所示，中间部分的绝对值之差如图6所示。

### 1.3.4 高斯核可分离实验

标准差为  $\sigma$  的高斯核可分解为两个标准差为  $\sigma$  一维高斯核的卷积，从而分解高斯核降低时间复杂度。实验结果如图7所示。

### 1.3.5 高斯核之差 DOG

将两个标准差不同的高斯函数  $G_{\sigma_1}, G_{\sigma_2}$ ，若有  $\sigma_1 < \sigma_2$ ，通过两个高斯核之差  $G_{\sigma_1} - G_{\sigma_2}$  设计 DOG 算子，然后下采样的到差分滤波核，窗口大小由  $\sigma_2$  确定。实验效果如图8和图9所示。

注：该部分不能使用彩色照片进行处理，因为涉及到图像做差，不同通道的之间的颜色无法对齐，最终结果会出现混乱。

### 1.3.6 图像锐化滤波器

$$\begin{aligned}
 f_{sharp} &= f + \alpha(f - f_{blur}) \\
 &= (1 + \alpha)I * f - \alpha G_{\sigma} * f \\
 &= ((1 + \alpha)I - \alpha G_{\sigma}) * f
 \end{aligned}$$

所以  $(1 + \alpha)I - \alpha G_{\sigma}$  为锐化的核,  $I$  为全通滤波器 (输入与输出图像相同), 例如  $3 \times 3$  的如下

$$I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

在锐化滤波器操作中, 由于做差后的图像正负值差距较大, 如果直接使用线性正规化方法处理溢出部分, 即:

$$f(m, n) \leftarrow \frac{f(m, n) - \min(f)}{\max(f) - \min(f)}$$

由于存在部分像素值较大, 而低像素值的像素偏多, 所以会导致整体色彩偏低. 为改进图像的亮度, 使用与 1 做最大值截断的方法进行正规化处理:

$$f(m, n) \leftarrow \begin{cases} 1, & f(m, n) - \min(f) > 1, \\ f(m, n - \min(f)), & \text{otherwise.} \end{cases}$$

执行相减后图像的亮度可能非常低, 需要手动提高亮度, 我将调节后的亮度均值维持在 0.5 附近, 这个调节亮度的操作就类似于锐化滤波器中的  $\alpha$ ,  $\alpha$  越大则最终图像的亮度越高. 两种正规化方法对比如图10所示.

从图11中可以看出  $\alpha = 1$  时锐化效果较好.

### 1.3.7 双边滤波

普通的高斯模糊只用到了图像像素之间的距离关系 (空域), 对每个像素使用相同的模糊处理, 所以在图像的边缘部分处理效果不好. 而双边滤波器通过加入像素值之间的关系 (值域), 从而能较好的对边界部分进行处理.

记图像为  $I$ ,  $\mathbf{p}, \mathbf{q}$  为像素点对应的向量,  $I_p$  表示图像中  $\mathbf{p}$  点对应的像素值,  $S$  为滤波器的向量空间,  $G_{\sigma}$  表示标准差为  $\sigma$  的高斯函数,  $\|\cdot\|$  表示 2-范数,  $W_p$  表示对滤波器进行归一化处理 (保持前后图像亮度一致), 则双边滤波器为

$$\begin{aligned}
 BF[I]_p &= \frac{1}{W_p} \sum_{q \in S} G_{\sigma_1}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_2}(|I_p - I_q|) I_q \\
 W_p &= \sum_{q \in S} G_{\sigma_1}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_2}(|I_p - I_q|) \quad (\text{归一化常数})
 \end{aligned}$$

由上述公式可知,  $G_{\sigma_1}(\|\mathbf{p}-\mathbf{q}\|)G_{\sigma_2}(|I_p-I_q|)$  为点  $\mathbf{p}$  处的双边滤波器核, 而  $G_{\sigma_1}(\|\mathbf{p}-\mathbf{q}\|)$  就是高斯核,  $G_{\sigma_2}(|I_p-I_q|)$  是值域之差作用高斯函数后的核, 两个核做内积即得到在点  $\mathbf{p}$  处的双边滤波器核, 然后进行归一化处理.

图12, 图13和图14提取了“黄鹤楼”照片中的塔顶, 屋檐和树边三处处局部双边滤波器核的效果图, 通过双边滤波器对比图, 即图15可发现, 当  $\sigma_r$  取值较大时, 双边滤波退化为 Gauss 模糊, 而且  $\sigma_r$  取值越小细节刻画更明显. 进一步, 实验了双边滤波对于降噪处理的效果, 如图16, 双边滤波祛斑美白的效果, 如图17.

### 1.3.8 傅里叶变换

一般的二维傅里叶变换公式为

$$\hat{f}(u, v) = \iint_{\mathbb{R}^2} f(x, y) e^{-2\pi i(\frac{ux}{M} + \frac{vy}{N})} dx dy$$

可以形象的理解为将图像  $f(x, y)$  向不同平面不同方向的复平面波做内积, 也即是求在  $e^{-2\pi i(ux+vy)}$  上的投影.

**可视化** 傅里叶变换所得到的的幅度谱和相位谱如图18.

注: 幅度谱输出先要取范数, 再取  $\log$  之后用线性变换将像素压缩到  $[0, 1]$  中进行图像输出.

将二维傅里叶变化后的空间称为  $K$  空间 (类似于信号处理中的频域), 则  $K$  空间中每一个像素表示一种二维正弦波, 则  $K_{u,v}$  处的辐角表示该种正弦波的相位大小,  $|K_{u,v}|$  表示该正弦波的幅度大小.

如果  $K$  空间的相位全部等于 0, 那么相当于平面波在相加的时候, 都没有移动, 所以图像一定会呈现一种周期性, 而且中间的点一点很亮. 这是因为复平面波没有移动, 那么所有的平面波在中心点相位为 0,  $\exp(0) = 1$ , 因此相当于所有的幅度叠加在一起了.

如果使得  $K$  空间的幅度全部等于 1, 那么相当于平面波在相加的时候, 只有移动, 而没有了各个波的大小信息, 低频成分和高频成分全都一样了, 所以图像应该比较嘈杂, 变化比较剧烈, 但是能看见大体的轮廓. (由于相位图逆变换结果只有少数像素值, 只能将像素值相对高一些的提高亮度, 否则特征十分不明显)

从图19中可以验证上述说法, 图20中交换了两张图片的幅度谱与相位谱, 可以更明显地看出相位包含了频域中主要的特征信息.

**频域滤波** 我们分别取  $K$  空间中低频部分 (也就是靠中间的部分) 和高频部分 (也就是靠近边缘的部分) 分别进行傅里叶逆变换, 实验效果如图21所示, 体现了低频部分包含了图像的主要信息, 而高频部分主要是细节与边界信息.

**幅度谱滤波:** 进一步, 对幅度谱进行 Gauss 滤波, 然后再复合相位谱, 最后进行傅里叶逆变换得到品与滤波后的效果, 如图22所示, 由于 Gauss 滤波使得低频与高频信息的幅度值减小, 从而不明显的细节部分被明显加强了.

频域滤波请见尾页修正部分.

## 1.4 实验步骤与结果分析

首先使用 PIL 库中的 open 函数进行图像的读入和灰度的转化，将像素值转化为  $[0, 1)$  之间的 numpy 中 ndarray 矩阵类，且图像矩阵统一保持三通道形状，便于后续处理，然后设计通用函数：图像归一化函数，并排绘图函数，制作 Gauss 核函数，零填充函数，卷积函数，并统一放置于 util 包中。

---

```

1 pi = np.pi # 计算高斯函数
2 def gauss(x, y, sigma, dim=2):
3     sigma_2 = sigma * sigma
4     if dim == 2:
5         return 1 / (2 * pi * sigma_2) * np.exp(-(x*x + y*y) / (2 * sigma_2))
6     return 1 / (np.sqrt(2 * pi) * sigma) * np.exp(-(x*x) / (2 * sigma_2))

```

---

在设计高斯核中，直接将高斯核大小设为  $2k + 1 \times 2k + 1$ ，标准差设置为  $\frac{k}{2}$ ，遵循  $2\sigma$  占比。

---

```

1 def gauss_filter(k, dim=2): # 返回一个 2k+1*2k+1 的 sigma=k 的高斯滤波器
2     n = 2 * k + 1
3     filter = np.zeros([n, n, 1])
4     for i in range(n):
5         for j in range(n):
6             if dim == 1 and i == k:
7                 filter[i, j, 0] = gauss(j-k, 0, sigma=k/2, dim=1)
8             elif dim == 2:
9                 filter[i, j, 0] = gauss(i-k, j-k, k/2)
10    return filter / np.sum(filter)

```

---

在 DOG 算法中，因为不同标准差高斯核大小不同，所以我没有将 DOG 滤波器计算出来，而是直接对图像进行相减做差，结果应该相同。

---

```

1 def DOG(img, sigma1, sigma2):
2     k1, k2 = np.ceil(sigma1 * 2).astype(int), np.ceil(sigma2 * 2).astype(int)
3     gauss1, gauss2 = gauss_filter(k1), gauss_filter(k2)
4     return conv(img, gauss1) - conv(img, gauss2)

```

---

在锐化处理中，因为最开始将像素值的数据类型设定为 np.uint8，导致图像直接相减得到的负数被自动向上取整为 0，所以使用卷积计算得到的结果与图像直接相减结果不同，亮度极低，后将数据类型统一为 np.float64，范围设定在  $[0, 1)$  中解决了该问题。

---

```

1 def sharp_filter(alpha, sigma):
2     k = np.ceil(sigma * 2).astype(int)
3     gauss = gauss_filter(k)
4     I = np.zeros([2*k+1, 2*k+1, 1])

```

---



```

5     I[k, k, 0] = 1
6     return (1 + alpha) * I - alpha * gauss

```

在双边滤波处理中，主要需要自己重新计算每个像素处的卷积核，所以需要重写一遍卷积，并在选定点处将卷积核和当前图像结果一并输出。

```

1 filter1 = gauss_filter(K)[: , : , 0] # 核大小为: a,b
2 for i in range(n): # 图像大小为: n,m,o
3     for j in range(m):
4         for k in range(o):
5             small_img = img[i:i+a, j:j+b, k]
6             gauss_line = lambda x: gauss(x, 0, sigma2, dim=1)
7             filter2 = gauss_line(small_img - img[i+K, j+K, k])
8             new_filter = filter1 * filter2
9             new_filter /= np.sum(new_filter) # 归一化
10            output[i, j, k] = np.sum(small_img * new_filter)

```

在傅里叶变换中，由于开始处理时忘记图像使用的是三通道，导致 `np.fft.fft2` 对三维矩阵进行操作频域图没有按照中心对称，去掉最后一维即可获得正确图像。频域分解为幅度谱和相位谱进行输出图片时，需要先使用 `np.abs` 转为实数，可能是因为幅度差距过大，如果直接使用线性正规化会导致图片亮度极低，所以需要幅度谱使用 `np.log`。

```

1 def fft(img):
2     img = img.reshape([img.shape[0], -1])
3     ft = np.fft.fft2(img)
4     ft_shift = np.fft.fftshift(ft)
5     ift = np.abs(np.fft.ifft2(ft))
6     magnitude = np.abs(ft_shift) # 幅度谱
7     phase = np.angle(ft_shift) # 相位谱
8     return [ft_shift, ift, magnitude, phase]

```

最后使用 `np.fft.ifft2` 进行逆变换后的结果同样是复数，所以需要使用 `np.abs` 转化为实数进行输出。

## 1.5 结论与讨论

通过本次实验，基本掌握滤波器的使用方法和图像的正规化方法，对于 python 中各种库的使用方法也有更多的理解，本次实验设计的通用函数处理包可以为以后更多实验提供便捷的操作。

## 1.6 主要代码及图像

由于代码量较大，完整代码请见[https://github.com/wty-yy/CVPR\\_homeworks/tree/master/code/hw1](https://github.com/wty-yy/CVPR_homeworks/tree/master/code/hw1)。



图 3: 零填充

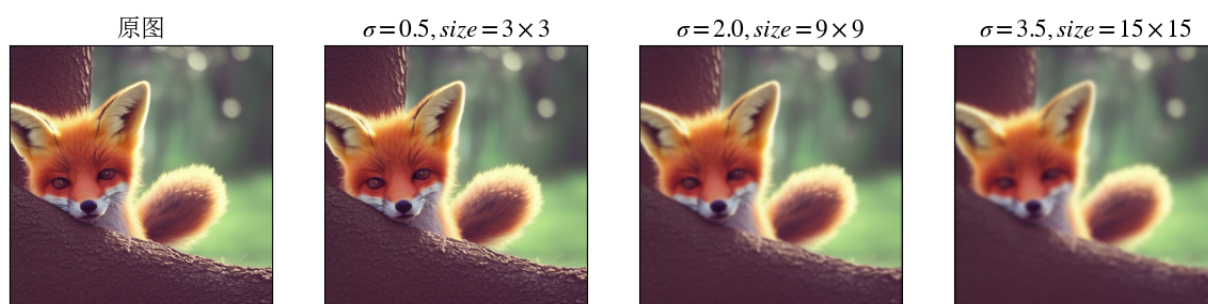


图 4: 高斯模糊效果

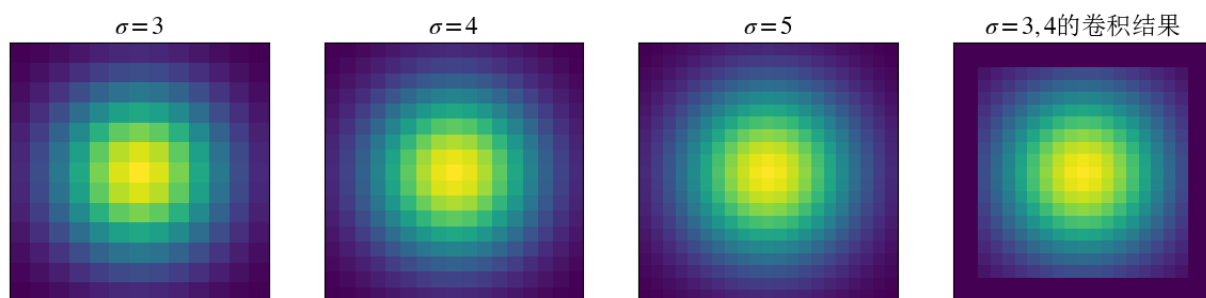


图 5: 高斯核卷积

```
[0.00022182 0.00024772 0.00025674 0.00024772]
[0.00024772 0.00027601 0.00028585 0.00027601]
[0.00025674 0.00028585 0.00029598 0.00028585]
[0.00024772 0.00027601 0.00028585 0.00027601]
```

图 6: 两个 Gauss 核卷积差值

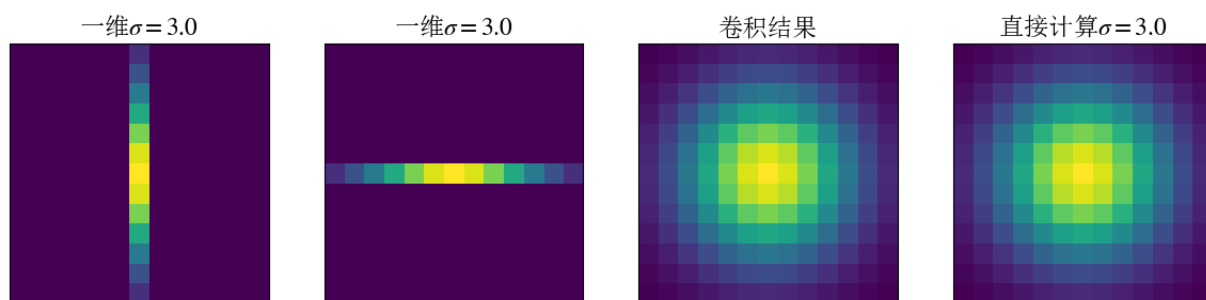


图 7: 高斯核分离

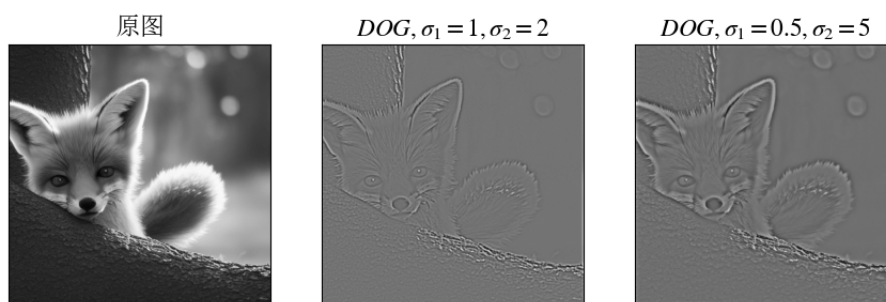


图 8: DOG 效果 1

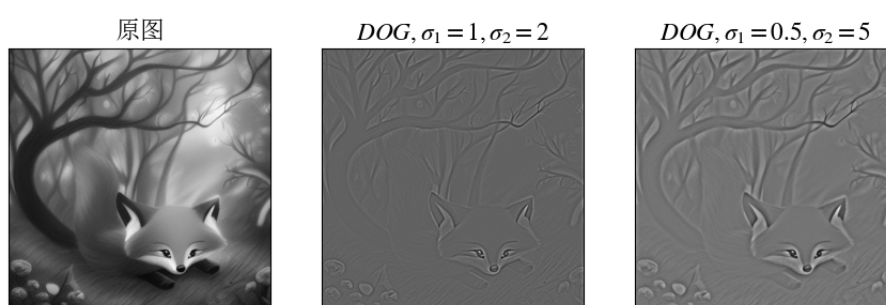


图 9: DOG 效果 2

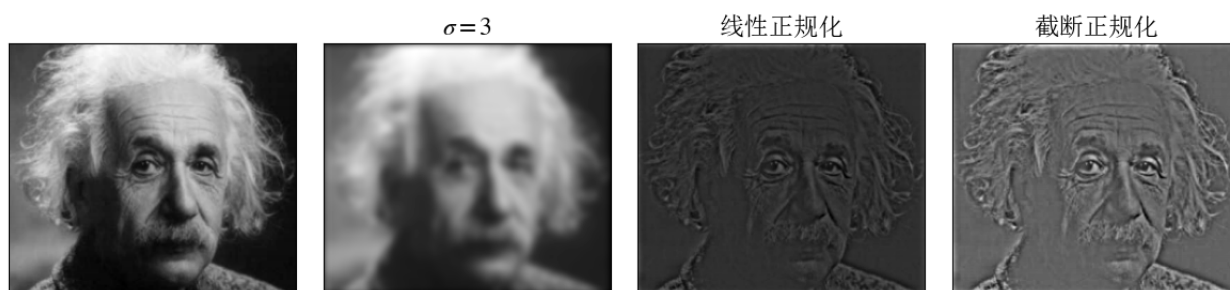


图 10: 线性正规化与截断正规化

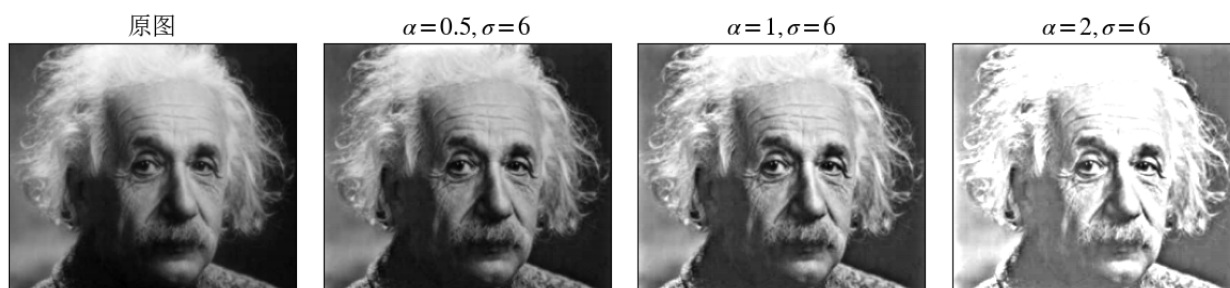


图 11: 锐化效果

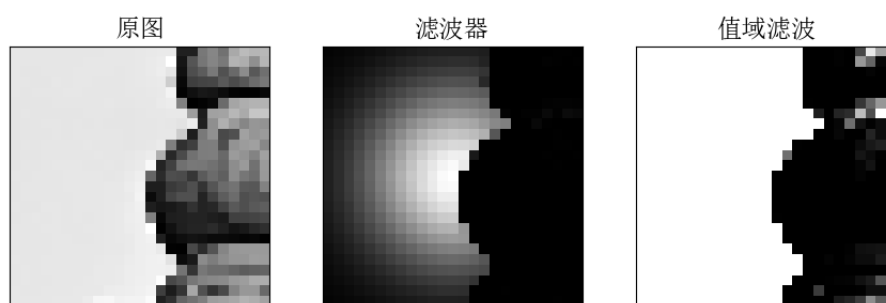


图 12: 局部滤波效果图 1

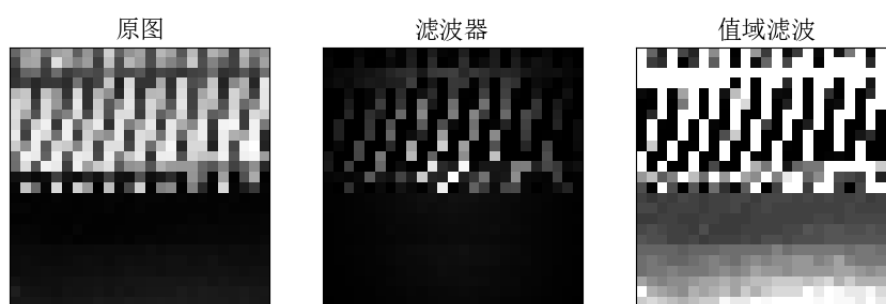


图 13: 局部滤波效果图 2

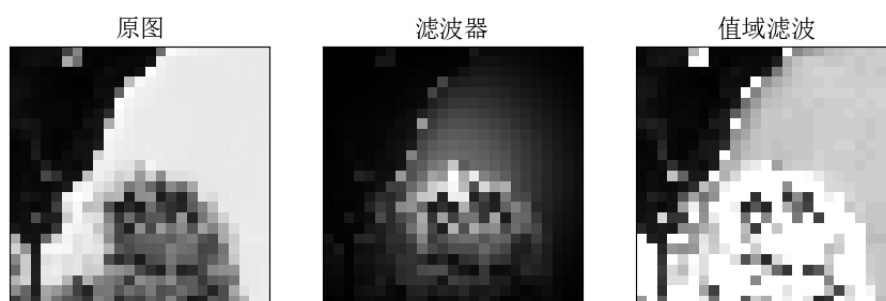


图 14: 局部滤波效果图 3

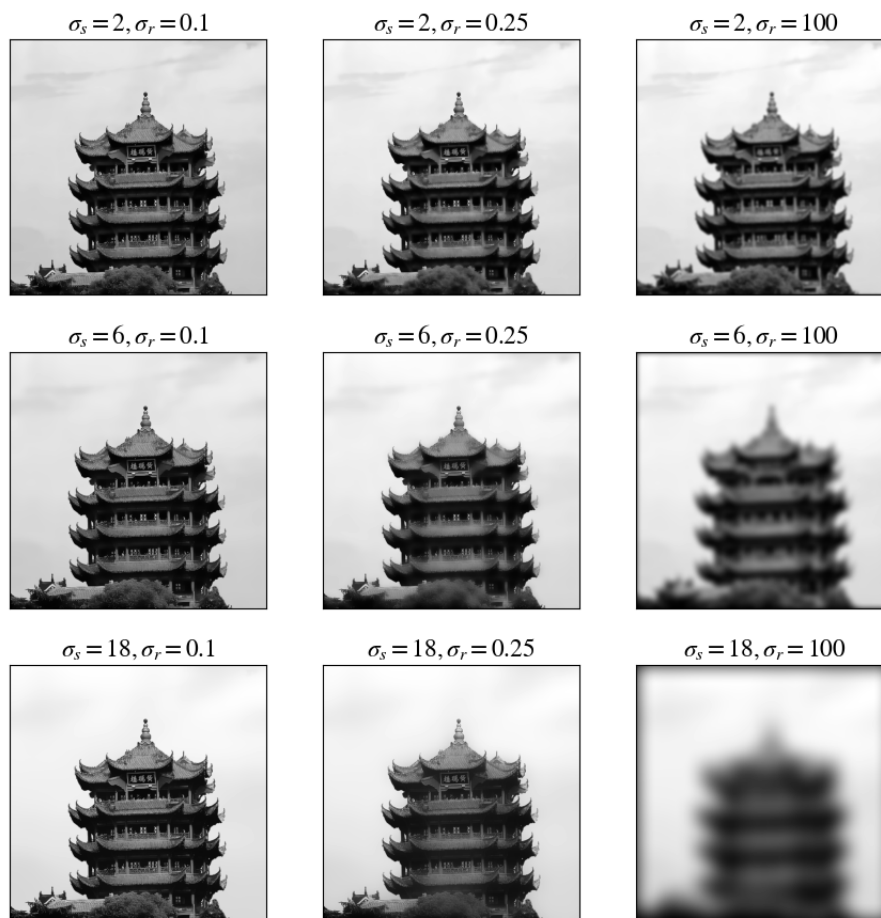


图 15: 双通道效果对比图

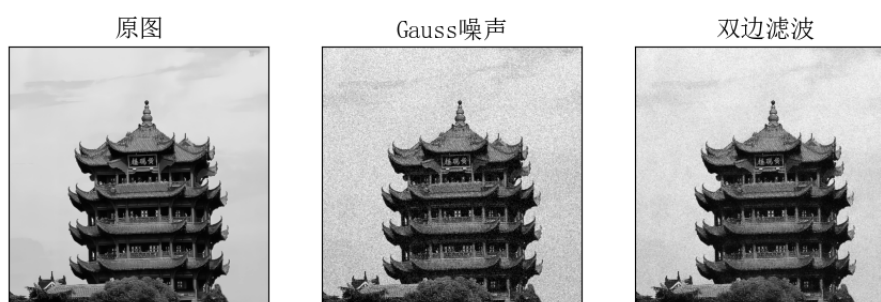


图 16: 双通道处理噪声



图 17: 双通道处理斑点

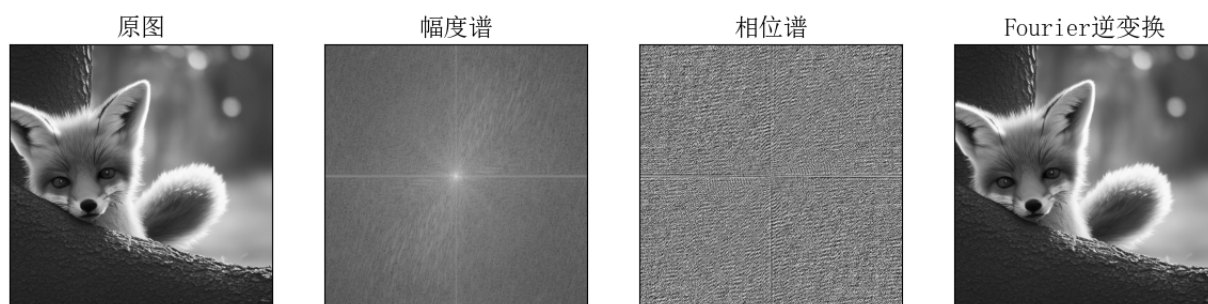


图 18: 幅度谱与相位谱

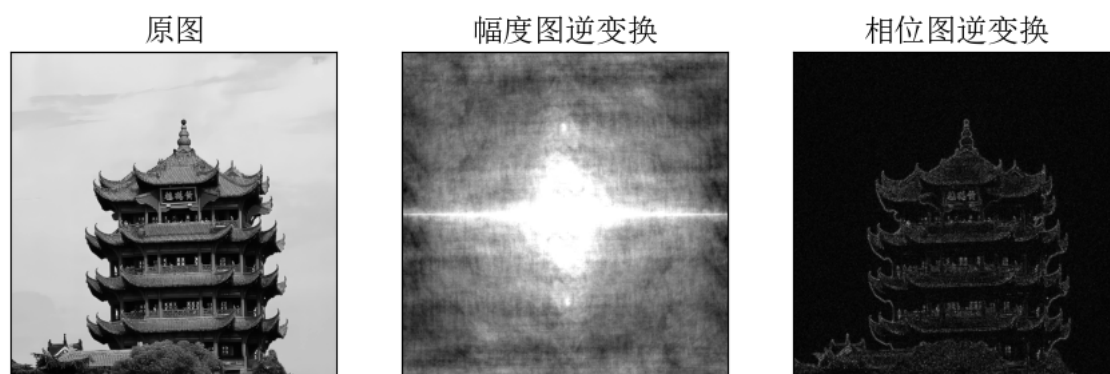


图 19: 单独分别做逆变换

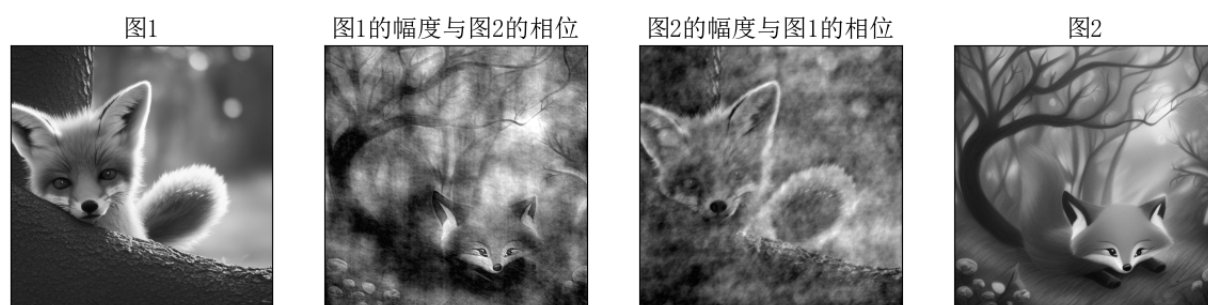


图 20: 交换幅度谱与相位谱

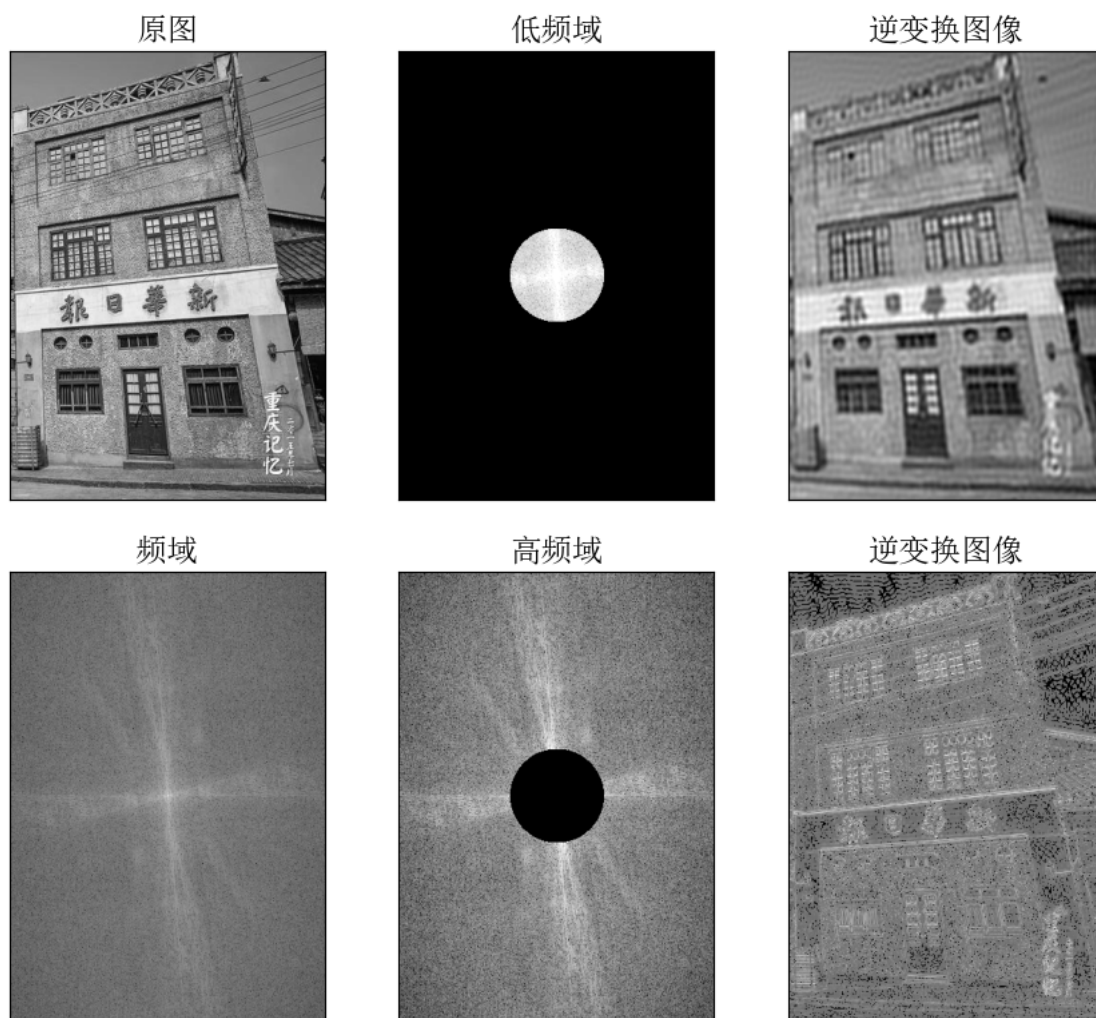


图 21: 低频域与高频与分离

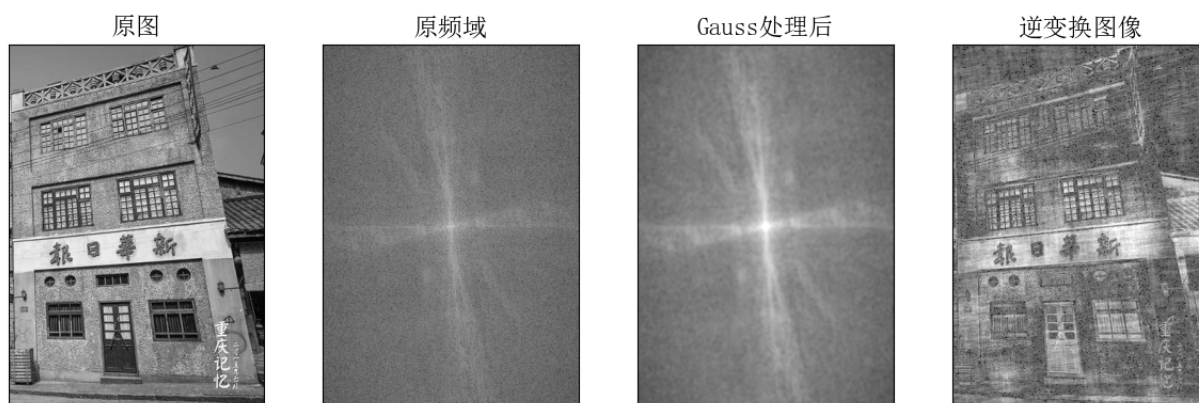


图 22: Gauss 处理幅度谱

## 频域滤波修正

使用大  $\sigma$  和频域做点积的结果，和使用小  $\sigma$  高斯核直接与图像做卷积得到的效果应该类似，突出细节（由理论推导最后一问可保证）

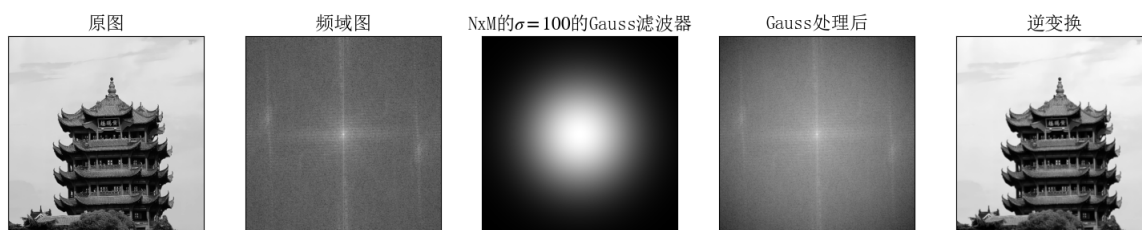


图 23: 频域滤波 1

使用小  $\sigma$  时注意输出频域图像时，不要再用  $\log$  进行调整了，因为像素已经相对非常集中了. 类似于使用大  $\sigma$  直接对图像进行卷积得到的效果，即高斯模糊.

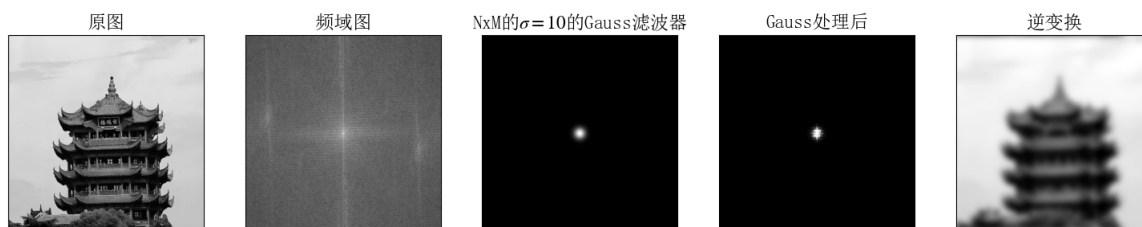


图 24: 频域滤波 2