# Name Convention

1. 变量符号及含义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 空域 | | 频域 | |
| 1D | 连续变量\* |  | ，采样过 |  | ，采样过 |
| 离散变量 |  |  |  |  |
| 2D | 连续变量 |  |  |  |  |
| 离散变量 |  |  |  |  |

\* Note, 这里只是要求连续变量(continuous variable), 不要求是连续函数（Continuous function, 这是和可微性联系在一起的，differentiability），如 unit impulse（亦称delta function， Dirac delta function就只是连续变量，不是连续函数）

1. 常用词汇

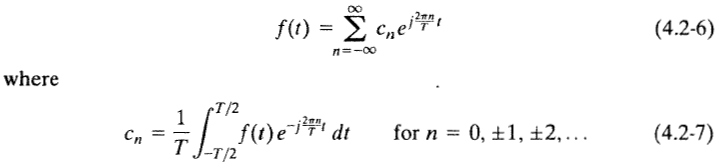
频谱 spectrum

幅值 Magnitude

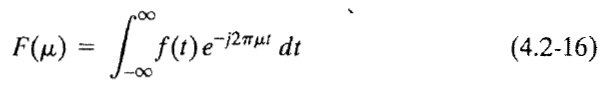
相位 Phase

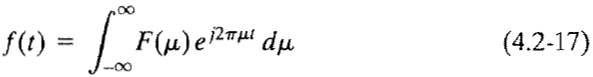
# 背景知识：傅立叶级数，变换与冲激

1. 傅立叶级数： 连续的周期函数f(x)，可以表述为 傅立叶级数加权的不同频率正弦余弦的和



1. 连续函数的傅立叶变换与反变换：





1. 单位冲激函数 unit impulse: continuous

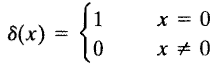
Machine generated alternative text:
00 
o 
if t = 0 
if t 0 

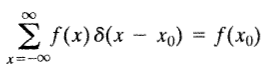
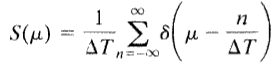
* 1. Sifting 取样
  2. Delta function 的傅立叶变换 恒等于1, 频谱为恒亮 (plate white)

这是 写成傅立叶逆变换的形式, 实际有，容易推导得到delta function的shift：

* 1. 由b. 函数和单位冲激的卷积，就是输入函数的一个拷贝
     1. 函数和移动到T的单位冲激的卷积，就是输入函数移动到T的一个拷贝
  2. 函数和单位冲激的相关，就是输入函数旋转180o后的一个拷贝

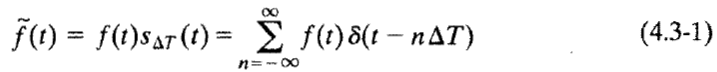
1. 单位离散冲激函数 unit discrete impulse:



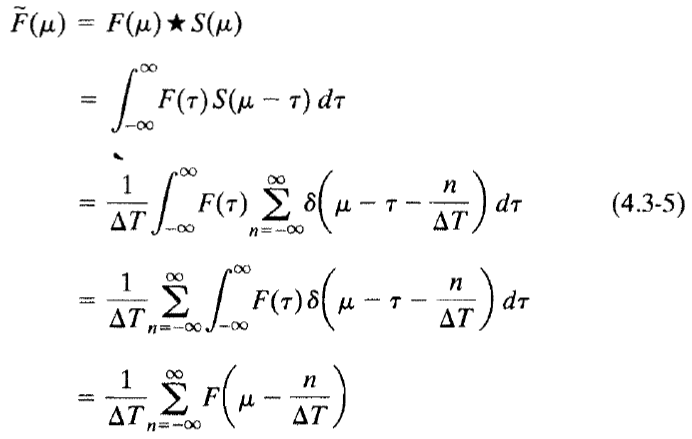
* 1. 离散的Sifting 取样
  2. 单位离散冲激串impulse train, 
  3. 单位离散冲激串的傅立叶变换仍然是 冲激串， 

# 无限长数字取样与复原

1. 数字取样的过程， 以一维连续函数为例
   1. 先看无限长度的数字取样，由连续变量的函数 乘以 单位离散冲激串 得到. ##只在 （k取整数，为取样间隔）处有取样值：



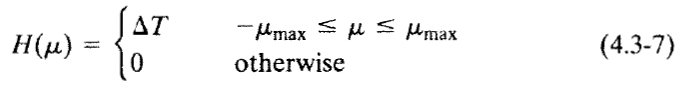
* 1. 再看无限长取样的频谱，由空域相乘即为频域卷积，及 单位离散冲激串 的傅立叶变换对公式 ## 下面公式表示 无限长取样的频谱是一个周期性函数，周期长度是



* 1. 取样down-sampling之后，如果我们想无损精确复原 原连续函数, 有两个要求：
     1. 取样频率 至少要能塞下一个完整的频谱 ，实函数的傅立叶变换的实部关于0中心对称的，虚部关于0负对称，因而若 频域正向最大频率为，则方向为，至此我们可得 取样辨率， 或取样间隔 .
        + 这就是著名的 “Nyquist-Shannon”采样定理。举个具体的例子，一个单一频率的正弦波 , 则采样间隔为 时，就能完整的复原这一正弦曲线
        + 对于一个函数，若其傅立叶变换，频率分布完全在之间，其余区段频率都为0，那么这样的函数被称为 **带限函数**（*band-limited*

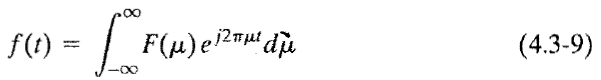
*Function*）

* + 1. 对于满足条件 c.i 的带限函数，且满足奈奎斯特采样定理的采样后，我们可以简单地通过一个频域低通滤波的矩形窗口函数，由 无限长取样的频谱 得出 连续函数的频谱 :

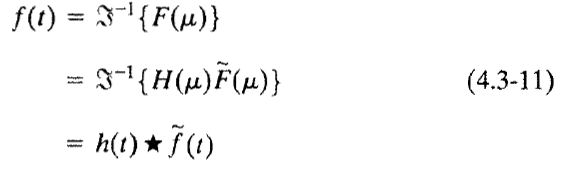


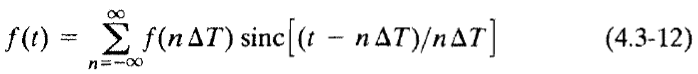


* 1. 最后一步，只要对 进行傅立叶逆变换就可以得到 原连续函数 或信号：



对上面五步进行合并简化，我们甚至可以直接由 无限长取样函数 得到 无限连续函数， 总结得出 up-sampling的公式:





可以从下面几点来理解公式（4.3-12）：

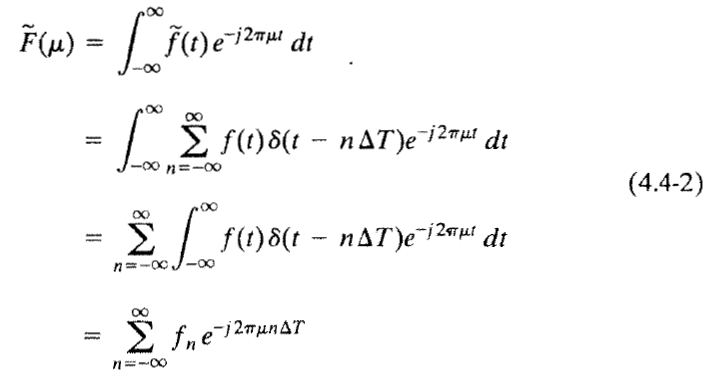
1. 无限连续函数 可以由 无限个等间隔的连续采样点 和sinc 函数的加权和，来无损完美复原
2. 因为的特性是 sinc(0)=1, sinc(k)=0| k为其他整数。所以可得，恰好是第k个取样之本身. 而 两个取样间隔中间的数, 如才是 采样点和sinc函数的加权和。

# 离散傅立叶变换DFT: 有限长的取样

## 1D DFT

从单变量的离散傅立叶变换来看这个问题

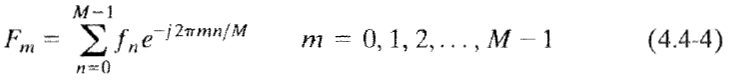
1. 由上节，见式（4.3-5），我们得出了无限长取样函数的频谱与无限长原函数频谱之间的关系，我们也可以直接 求与无限长离散采样 ，乃至空域原连续函数之间的关系。只要将 描述的（4.3-1）式带入傅立叶变换公式（4.2-16）：



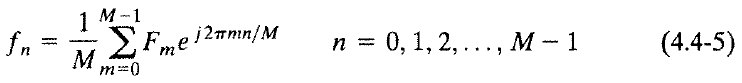
1. 还是由（4.3-5）式，我们只需要一个周期的频谱 ， 而且我们在频域也均匀采样下(频域在点上采样也使得DFT形式上与 无关)：



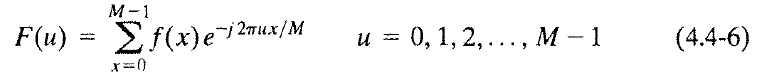
有限离散点对应的 傅立叶变换 记为 ,将带入式（4.4-2）可得，离散傅立叶变换集

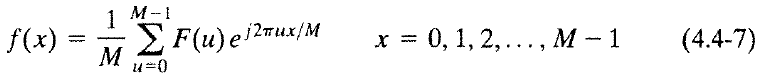


这就是一维 有限长的 离散傅立叶变换 1D DFT，由带入傅立叶反变换公式，我们可以得到均匀采样的有限长的离散采样集：



1. 最后用文章开头的 name convention来改写上面两式，n => x, m => u得到一维 DFT对：

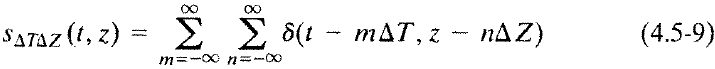




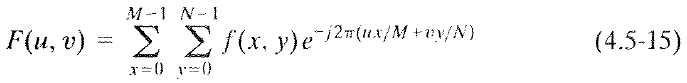
## 2D DFT

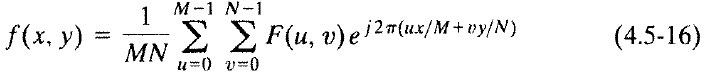
将1D DFT 扩展到2D：

1. 二维单位离散冲激串，同有Nyquist采样频率要求 ，



1. 二维离散傅立叶变换集 2D DFT :

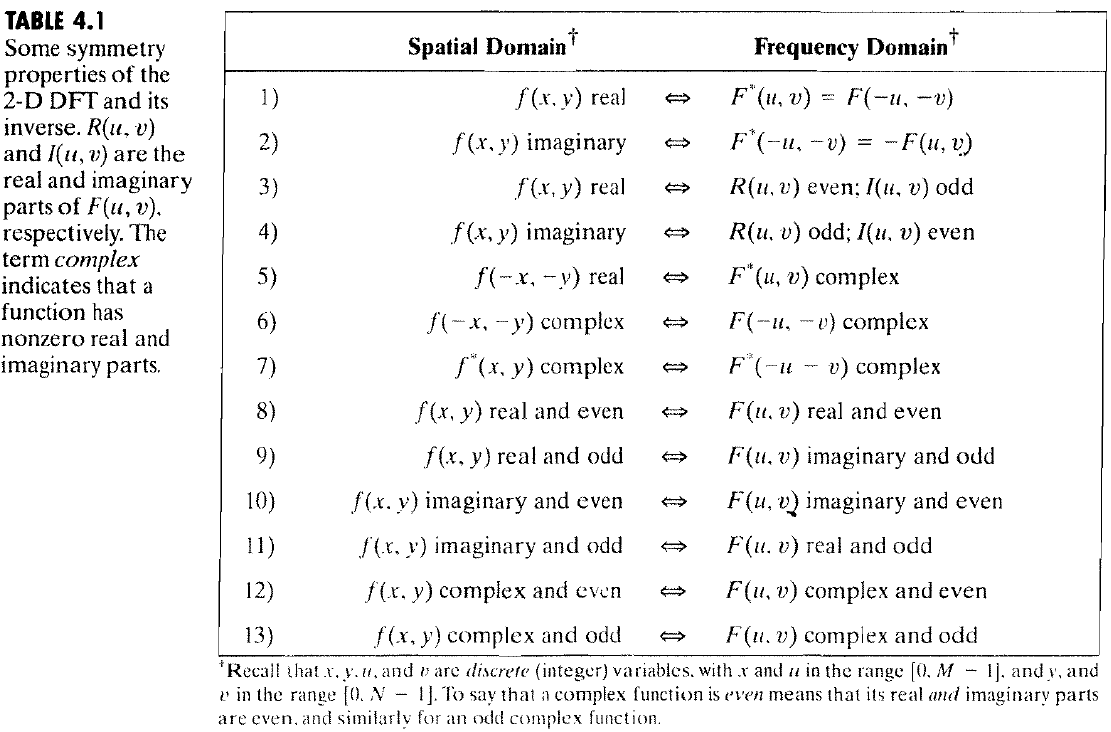




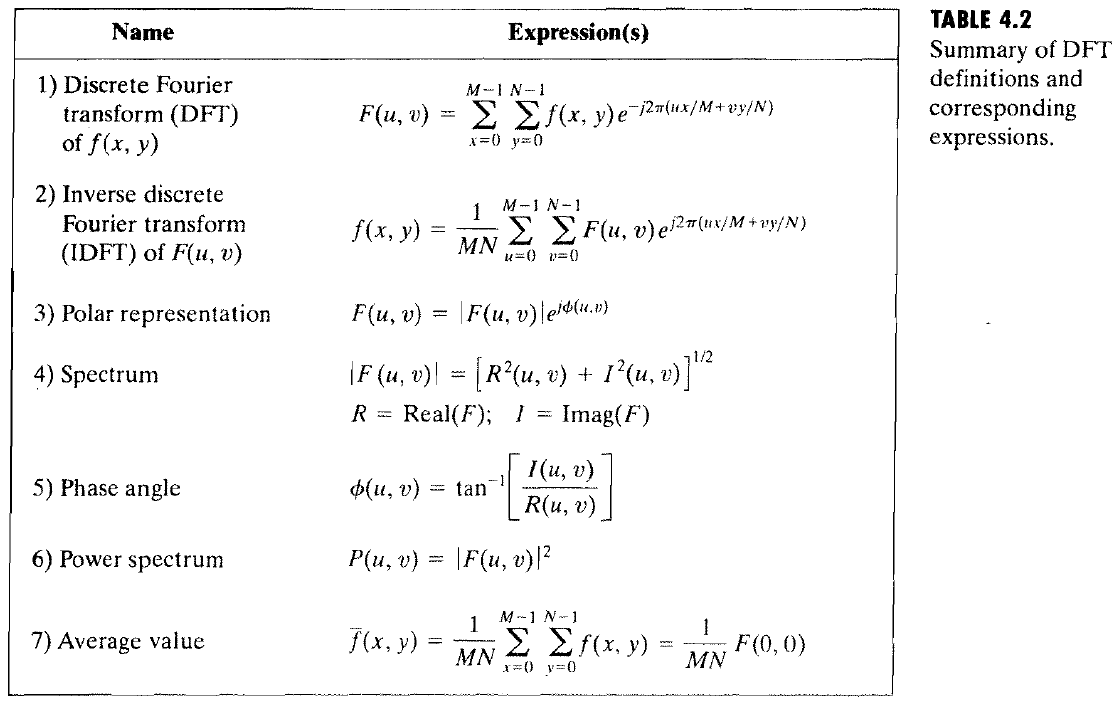
其中输入的image f(x, y) size为MxN.

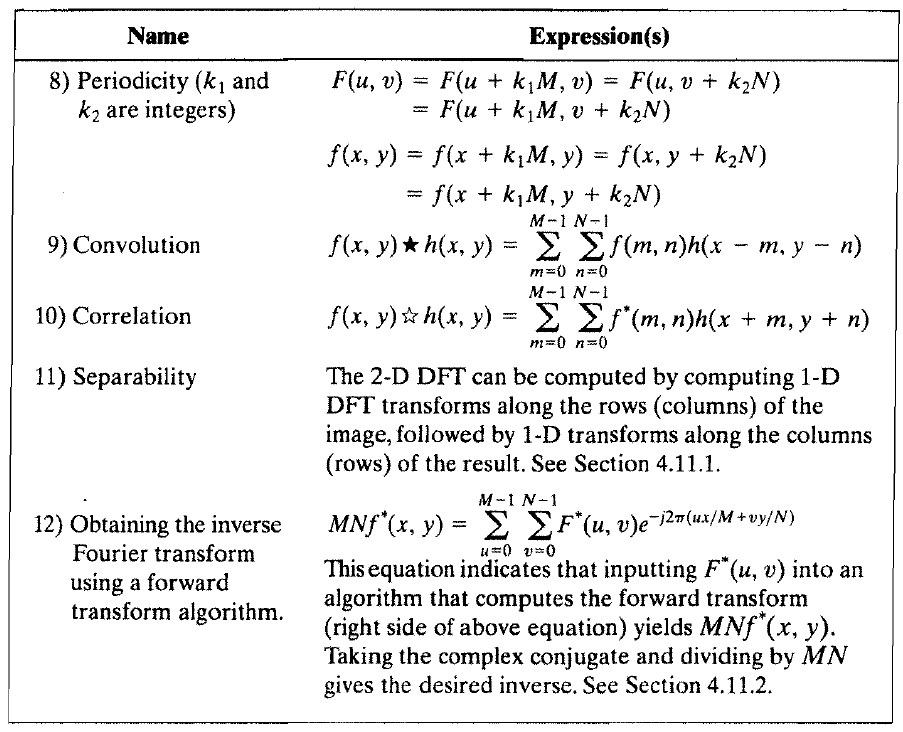
## DFT property summary tables

### Table 1. symmetry in DFT

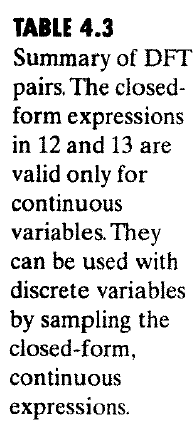
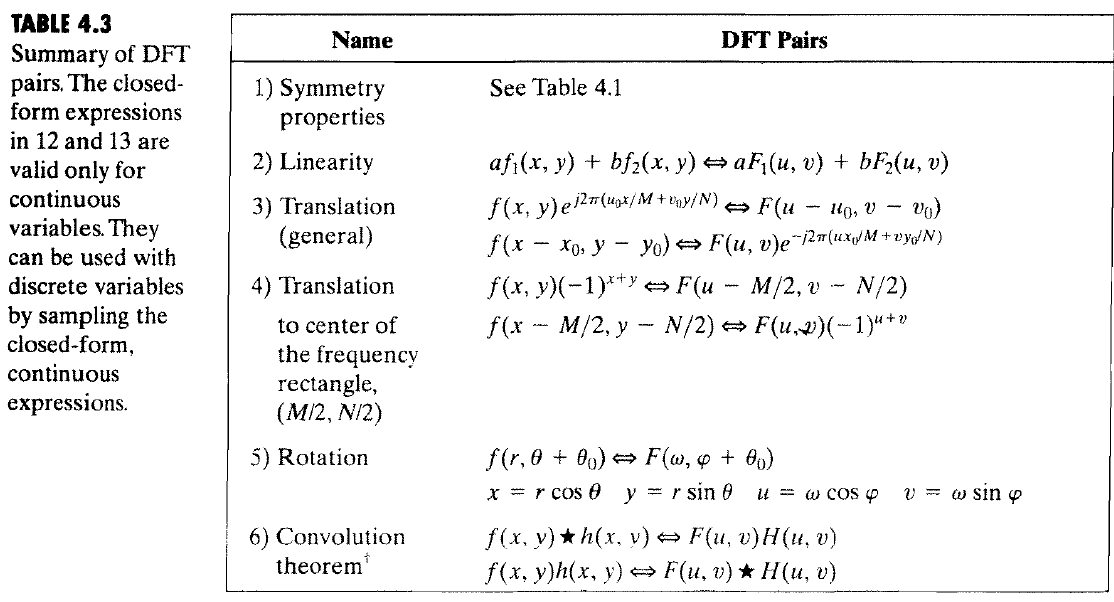


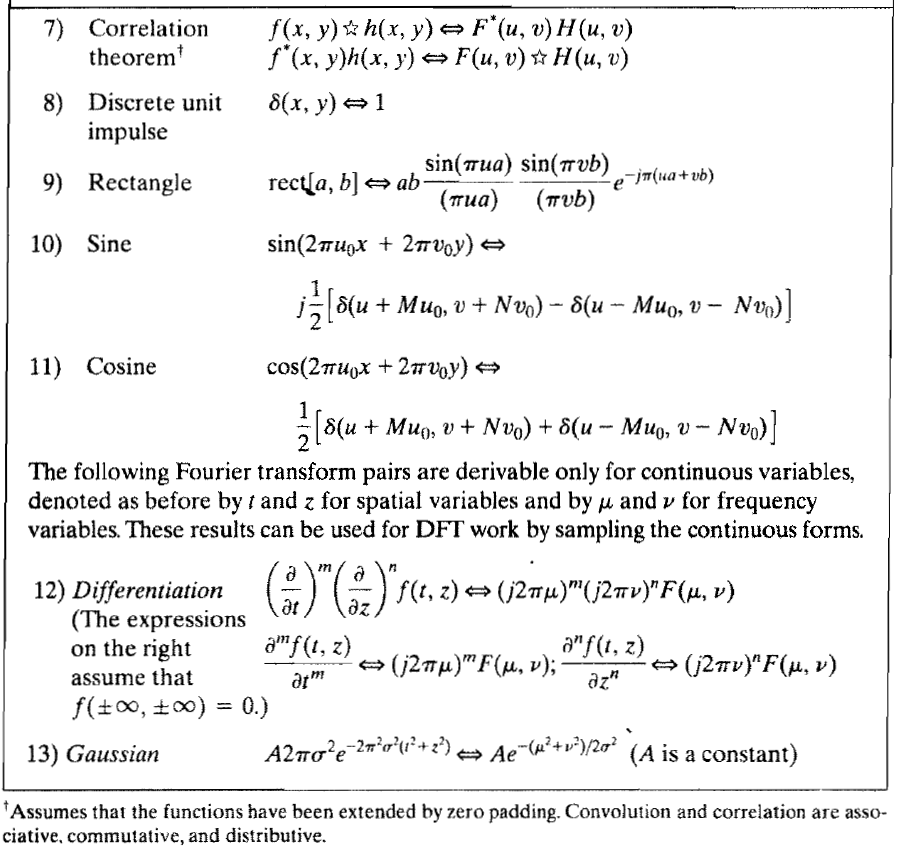
### Table 2. DFT definition and related expressions





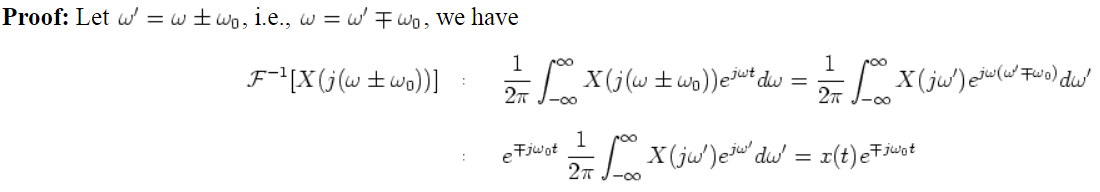
### Table 3. DFT Pairs





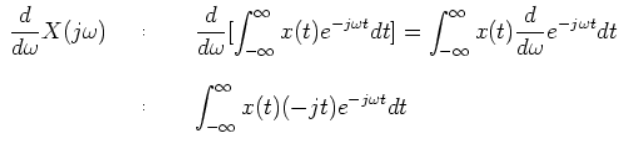
### Proof for **Frequency shift / translation**

Table 4.3 rule 3).general translation Proof



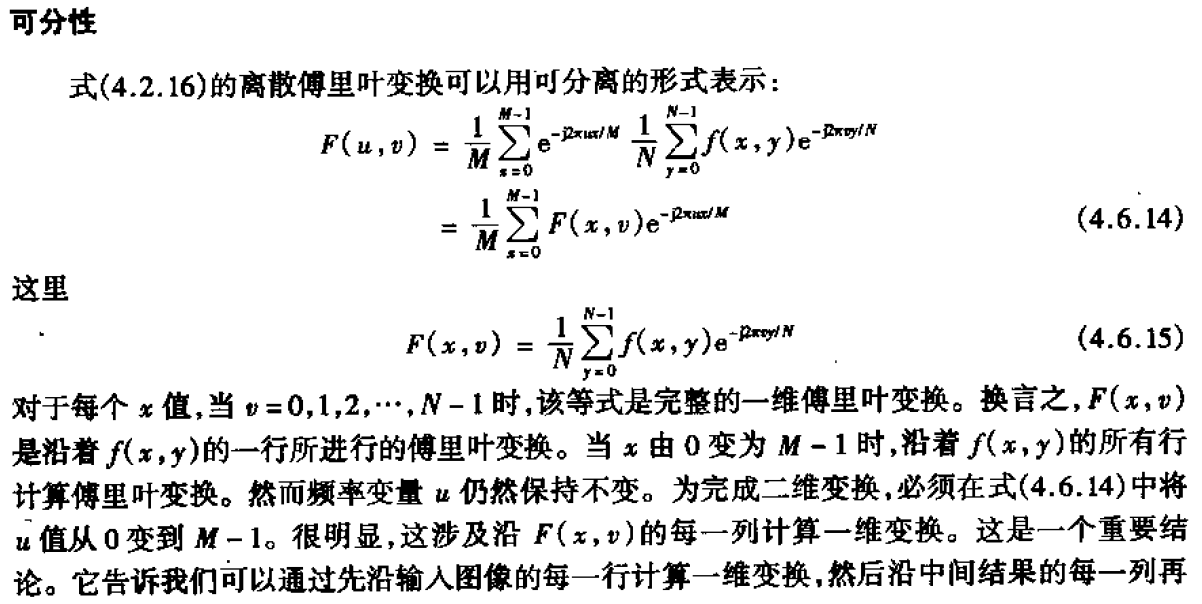
### Proof for **Differentiation**

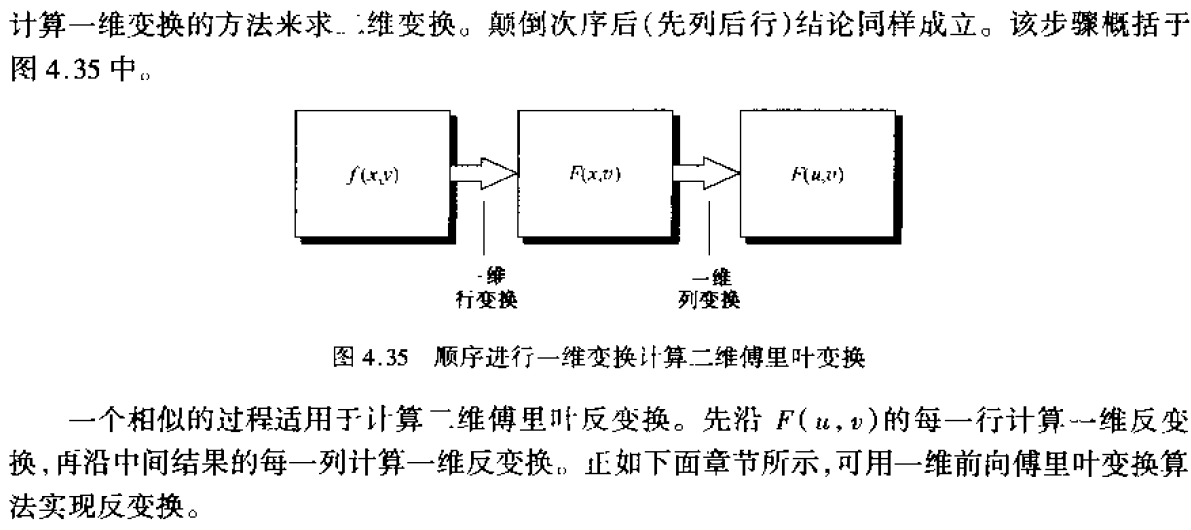
Table 4.3 rule 12) Differentiation Proof:



### Proof for **DFT Separability**

Table 4.2 rule 11) Separability for **DFT** Proof:





### Proof for **Convolution Separability**

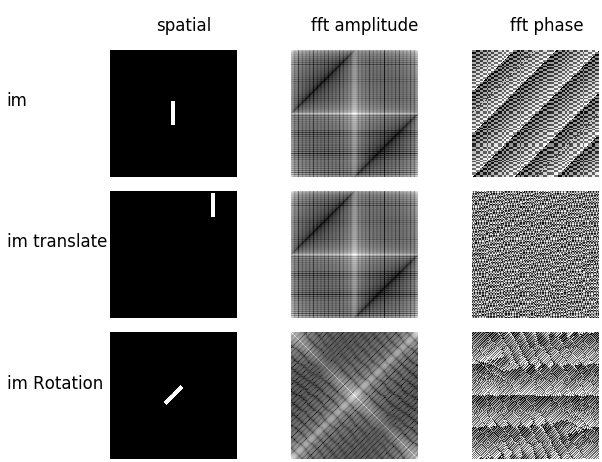
Separability for Convolution Proof:

Machine generated alternative text:
UllR*JPDvJsYy3x3, 
191] , 
ifij]l, 
(WI) (IxN) 
B. [abc 
[a 

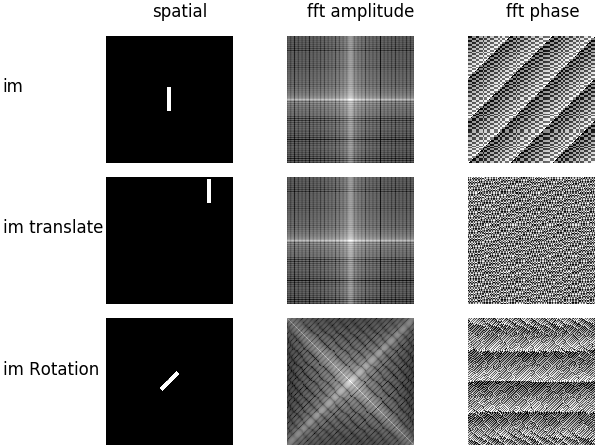
Machine generated alternative text:
jüÄ2D.=-X , 
y[m, n 
x[m, n] * h[m, n] 
y[m, n] h[m, n] * x[m, n] — 
[m, (Mxl) , 
mm, n] = [m] • 112 [n] 
hill [m, n]fVOjE, 
XP, j] • h[m i, n j] 
h[i,j] • i, n j] 
hi [i] • h2[j] • — i, n — j] 
Ill [i] • x[m, — i, n —j] 

## Implementation for how translation/rotation effect DFT

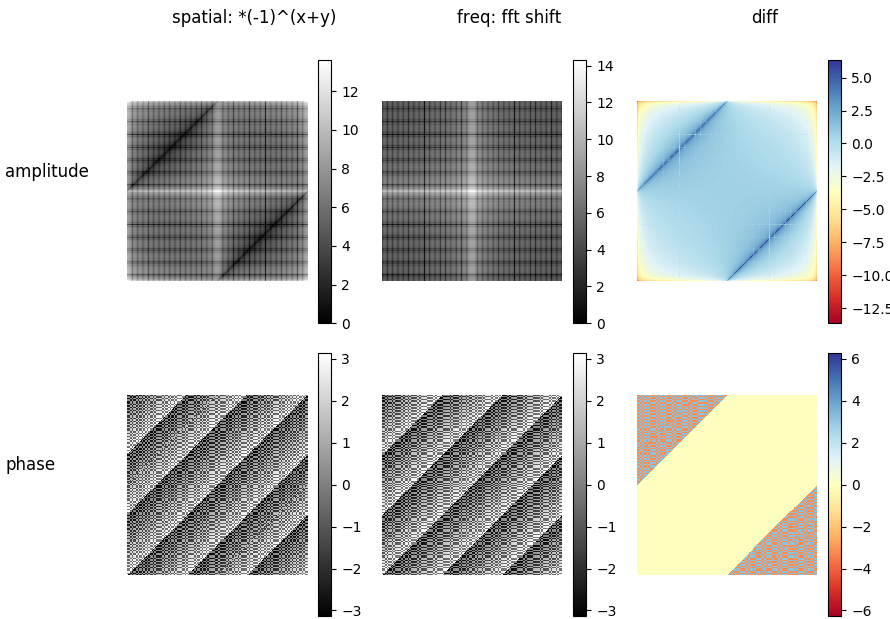
* When to get FFT after multiply (-1)^(x+y):



* If get FFT, use np.fft.fftshift

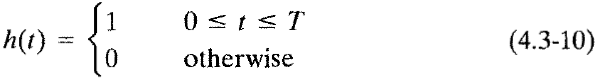


The FFT difference on the original Image



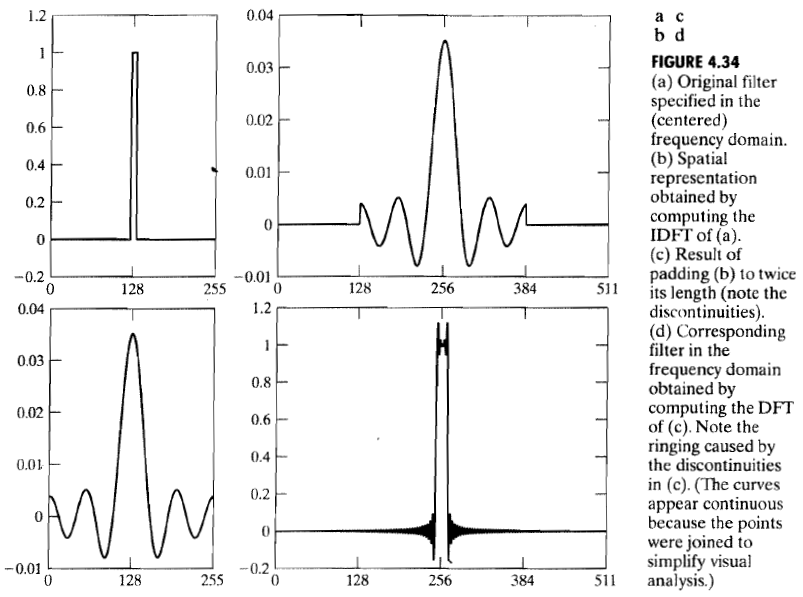
## Personal Tips for DFT

1. 图像混淆aliasing：
   1. 空间混淆: 来源于采样时的频率混淆（DIP E3 cn, P135, 4.3.4）
      1. 频率混淆（frequency aliasing）: 是指对一个带限函数以低于奈奎斯特采样率采样（即欠采样），最终效果是频域周期重叠，相邻周期混淆在一起，无论怎样滤波都无法得到单周期，也就无法还原出原来的信号。
      2. Issue Description: 频率混淆在真实信号处理场景中无法避免，因为即使原函数真为带限函数，但是采集信号时一定是有限时间（或空间的），这样原带限函数需要乘以一个时间（或空间）区间函数：



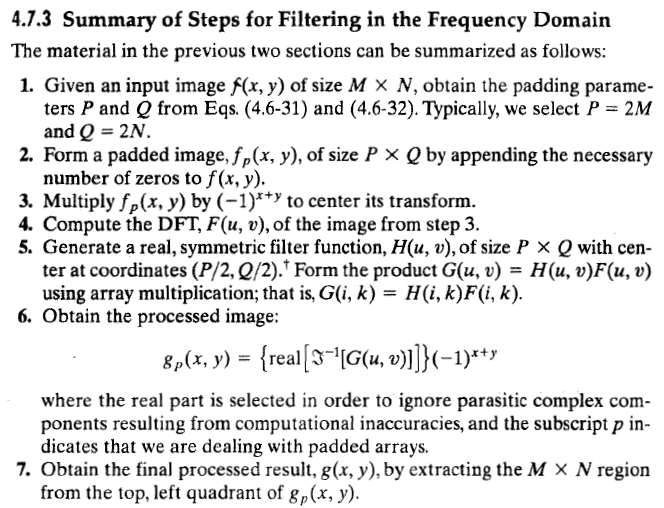
* + 1. 这样在频域的函数就不是带限函数，因为盒状函数的傅立叶变换是一个频率分量无限宽的函数（从到），据卷积定理，, 第一个带限函数卷积上一个频率分量无限宽函数，还是一个频率分量无限宽函数，这样就不可避免会有“频率混淆”。
    2. Solution：不过我们可以在采样之前，就对连续函数平滑，减少高频分量，来“抗混淆”, 具体硬件方法比如“适度离焦采集图像defocusing”
  1. 时间混淆： 于图像序列之间的时间间隔有关，如车轮效应，电视画面中当汽车在前进时，车轮却在反转的现象。序列图像帧率低于轮子的转动速度造成的。（这里不是重点）

1. DFT卷积时的**纠缠错误** wraparound error
   1. Issue Description:（DIP E3 cn, P157, 4.6.6）. DFT卷积时，我们必须考虑DFT表达式固有的一个频域周期性。很容易出现的错误是，翻转的卷积函数滑过了过去周期的原函数( the flipped (rotated) version of to slide completely past ), 这样会导致错误的结果，就叫DFT卷积“纠缠错误”
   2. Solution: 不过所幸的是纠缠错误，可以通过简单的 零填充（zero padding） 有效的解决。
2. 频率泄漏*frequency leakage*
   1. 这是用零填充解决DFT卷积时的纠缠错误时，所带来的side effect
   2. Issue Description: 如果卷积的双方双方有一个采样区域的尾端不是0，而我们再用零填充方法，给函数尾端填0，这是这个函数在填充junction处就是一个突变，空域来看是乘以了一个“盒状”函数，频域来看这次padding就是相当于卷积上了一个sinc函数，这样就会导致sinc函数高频分量导致的频率泄漏，在图像上也会产生块效应（blocky effect）。
   3. Solution：Firstly， leakage never can be totally eliminated，but it can be reduced significantly by multiplying the sampled function by another function that tapers（attenuate衰减） smoothly to near zero at both ends of the sampled record（取样数据） to dampen (decrease, 削弱) the sharp transitions (and thus the high frequency components) of the box. 这种方法就叫做 **开窗**（*windowing*）或**切趾**（*apodizing*）
3. 振铃效应（Ringing artifacts）
   1. Issue Description:  图像处理中，对一幅图像进行滤波处理，若选用的频域滤波器具有陡峭的变化，则会使滤波图像产生“振铃”，所谓“振铃”，就是指输出图像的灰度剧烈变化处产生的震荡，就好像钟被敲击后产生的空气震荡。 Ringing artifacts are artifacts that appear as spurious signals near sharp transitions in a signal, visually, they appear as bands or "ghosts" near edges.
   2. LPF为例：振铃效应(Ringing Effect)



* 1. Solution： 当然我们可以说上面的例子可以通过在 零交叉点来截断 LPF经iDFT得到的空域函数，再做 0填充，就能保持连续性。不过我们需要更一般性的solution，我们的目标是：work with specified filter shapes in frequency domain(including ideal filters) without having to be concerned with the truncation issues. 一种方法是：先zero-padding images, 然后在频域创造和padded image一样size的filter，当然这样的话，因为原函数和filter中有一项filter没有padding会导致我们所说的第2种错误：DFT卷积时的**纠缠错误**。不过，在实践中，这种错误可以通过image zero-padding 提供的间隔有效地减轻（In practice, this error is mitigated significantly by the separation provided by the padding of the image, and its preferable to the ringing artifact.）。而平滑型的滤波器这类问题更少。

# 频域滤波



## 高斯滤波器Gaussian Filter

GLPF





HPF as the difference of Gaussians

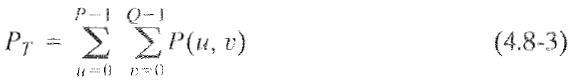
 

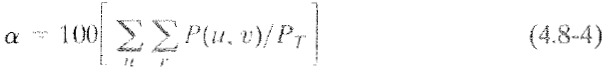


Sobel operator in Frequency domain

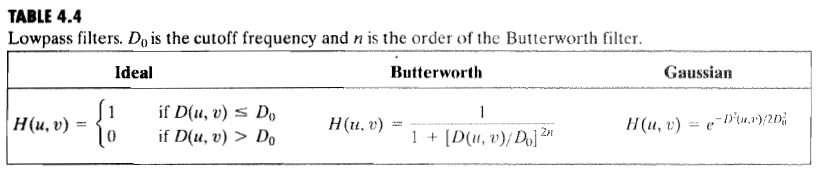
## 低通滤波器 Low Pass Filter

表达LPF 截止频率圆内的傅立叶功率集中度的方式：

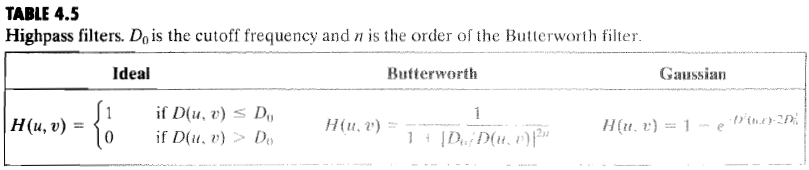




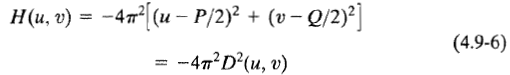
Size P, Q都是padding之后的image size



## 高通滤波器 High Pass Filter



## 拉普拉斯滤波器Laplace Filter





## 退化模板Unsharp Masking, 高提升滤波HighBoost Filtering, and 高频强调High-Emphasis Filtering







更一般的形式：





## 同态滤波 Homomorphic Filtering

Homomorphic Filter, Based on the illumination-reflection mode, consider an image as the product of illumination component(which usually is slow spatial variation) and reflection component (which usually is the big vary esp the junction between different objects).

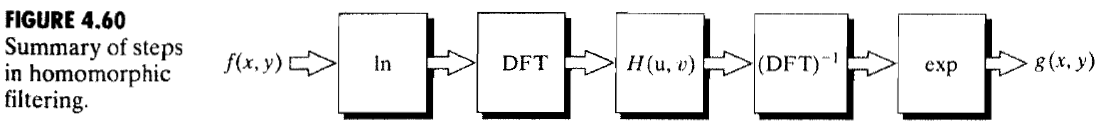
f(x, y) = i(x, y) \* r(x, y)

Homomorphic Filter (下式由高斯高通滤波器GHPF构建):



虽然上面公式看上去和 高频强调滤波器 类似，γL 是全频谱的权重（对应照明分量+反射分量），γH – γL是 GHPF的权重（主要是反射分量）

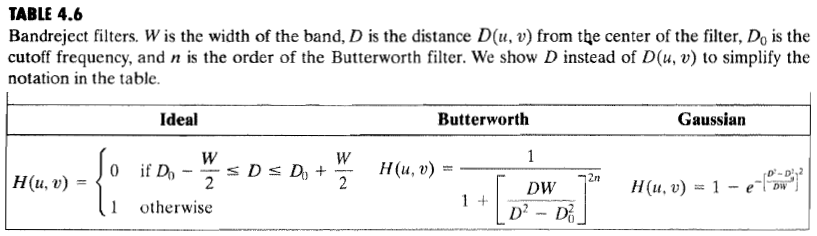
但是，其同态滤波的过程也和其他频域滤波器的过程完全不同：



而且，如果image 在0-255灰度值范围内，真有点的强度值为0的话，应将image 强度值全加1， 以免出现 ln(0)的情况，而且滤波完成后要记得再减去1还原图像。

## 选择滤波 Selective Filtering

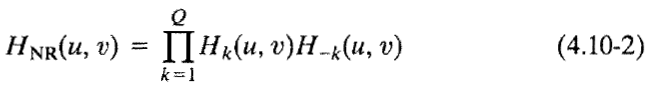
### 带阻 Band Reject Filter



### 带通Band Pass Filter



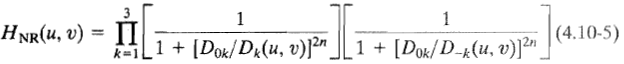
### 陷波带阻 Notch Band Filter



Where 是以为中心的高通滤波器，是以为中心的高通滤波器，非原点为中心的滤波器必须成对出现，才能满足“zero phase shift filter”的要求。







从高通滤波器的角度来理解陷波带阻滤波器，高通滤波器是中心为0，D0外为1，这里 中心 处恰为0，边界截止频率 可以相同，也可不同，以中心画半径为的圆，圆外频率通过。

### 陷波带通 Notch Reject Filter



## Implementation

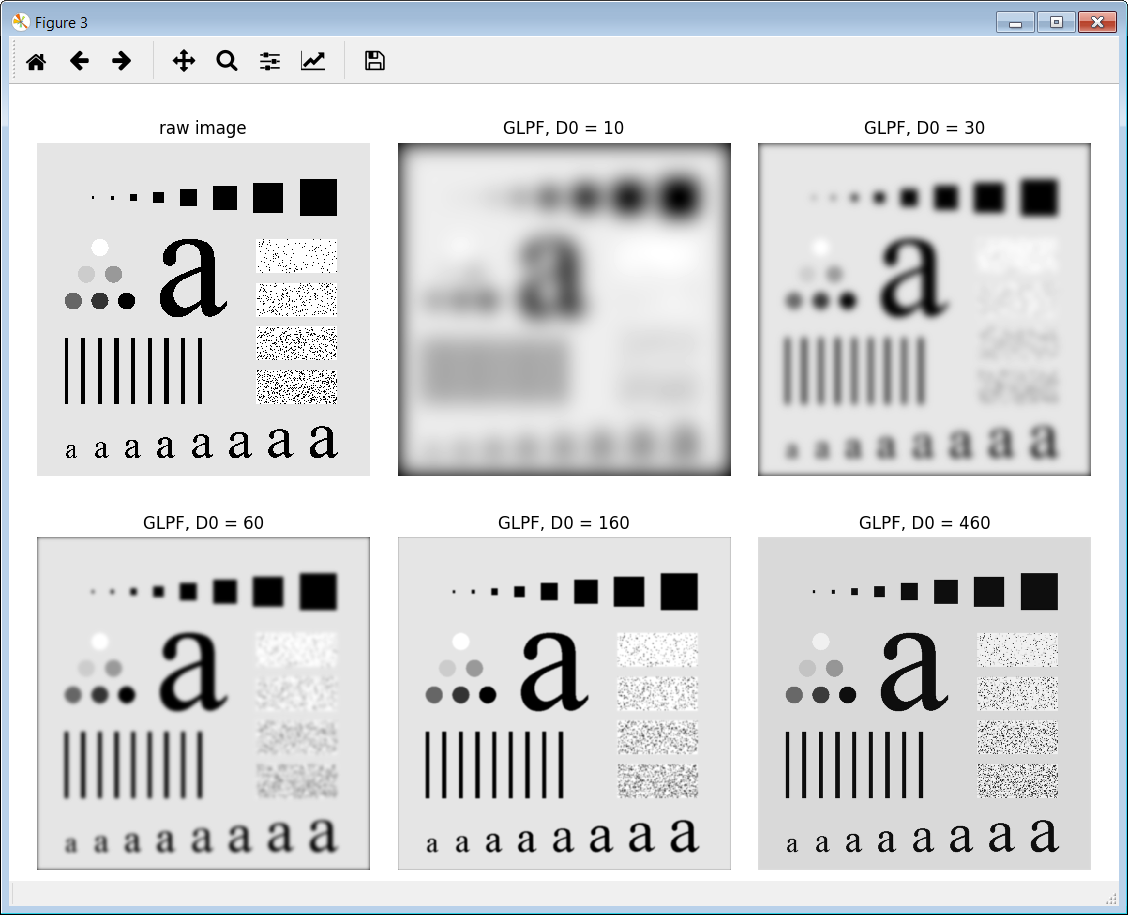
1. ILPF明显的“振铃效应”：



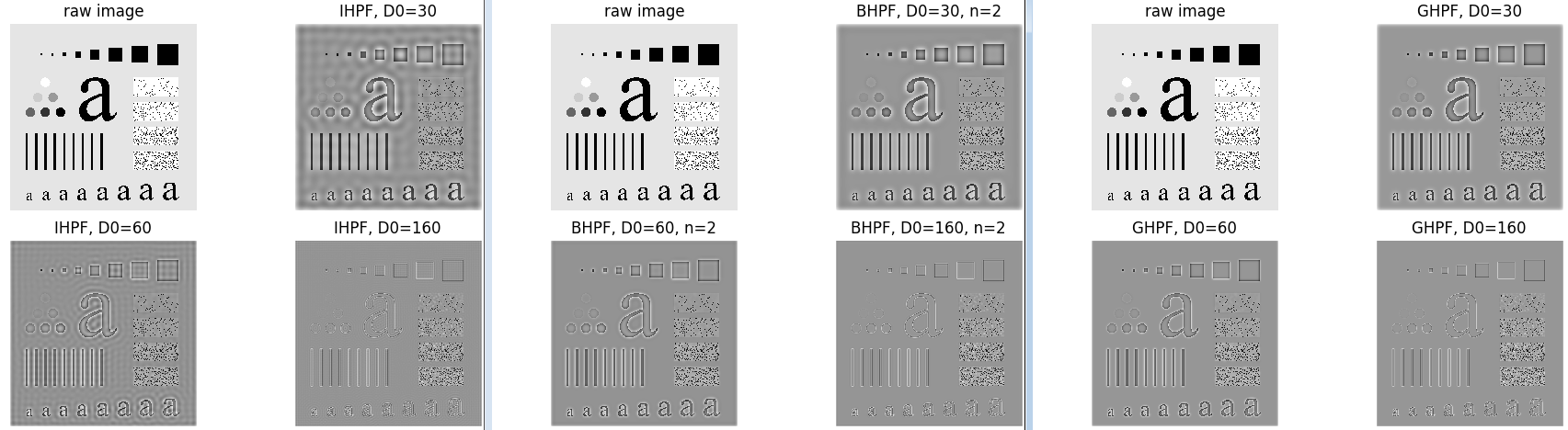
1. BLPF, 可以看出n=2, 二阶的布特沃斯低通滤波器，没有明显的“振铃效应”，但是平滑的效果也比ILPF差一点。事实上，当n 较小时， 布特沃斯低通滤波器接近于高斯低通滤波（特别是当BLPF | n=2，GLPF和BLPF的泰勒展开项相同）



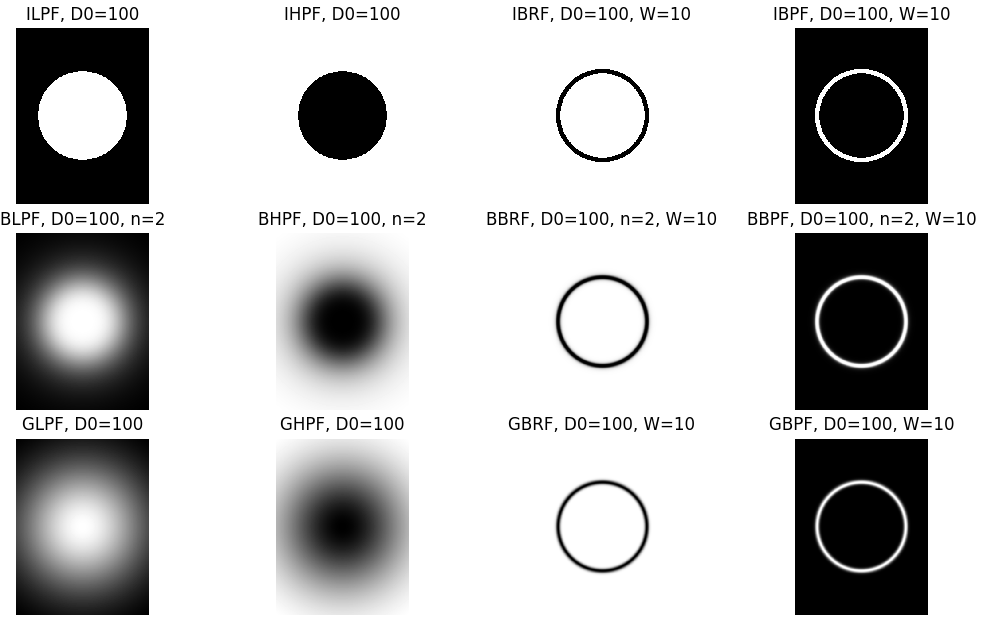
1. GLPF



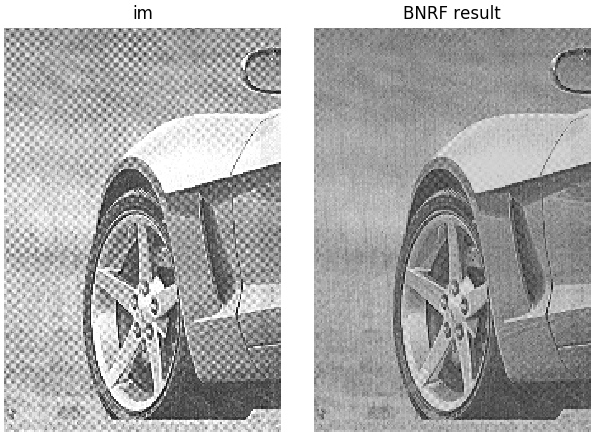
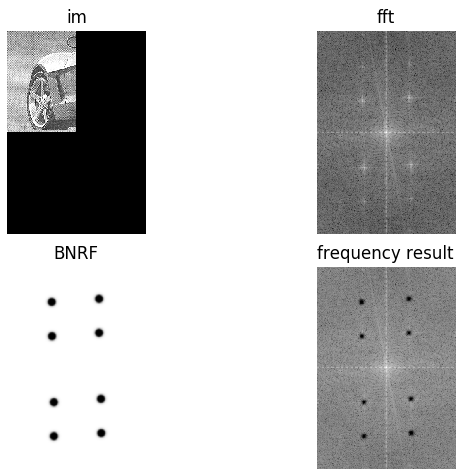
1. HPF



1. Visualize all the filters, shape(400, 300)



1. Notch Reject Filter



# Appendix A: Dirac and Related Functions

https://www.zhihu.com/question/36593219

