## 图像处理与模式识别技术报告

本次大作业主要实现了灰度图像的快速傅里叶变换与逆变换，离散余弦变换与逆变换，在此基础上还完成了低通滤波器与高通滤波器各三种。下面分条进行详细描述。

**快速傅里叶变换与逆变换**

**算法分析**

使用傅里叶变换与逆变换，我们可以将图像的空间域与频率域联系起来，从而对图像完成一些从空间域无法实现的处理。然而，原始的一维傅里叶变换具有O(n^2)的复杂度，而在实际中需要处理的n往往较大，如此高的时间复杂度是无法接受的。因此，在实际工程中，需要使用复杂度为O(nlogn)的快速傅里叶变换。

**一维傅里叶变换**

一维傅里叶变换将信号从时间域转换到频率域，其公式如下：



其中，X(k)为信号的频率域表示，x(n)为频率的时间域表示，。

注意到WN有以下三个特点：

1. 周期性：



1. 对称性



1. 退化



而利用这两点性质，则可省掉一些不必要的计算，对傅里叶变换进行加速，下面进行详细阐述。

对序列x(n)取N点DFT，假定N是2的整数次方，N = 2M。













从而将N点的DFT转化为了两个N / 2点的DFT之和。注意上面的公式中，两个小规模的DFT只能算出k = 0, 1, ..., (N - 1) / 2的值，而另外一半的值可以利用周期性与对称性获得，即:

 ①

 ②

 ③

综合两部分结果，我们得到

 ④

 ⑤

其中k = 0, 1, ..., (N - 1) / 2。

由此，我们便将N个元素的DFT转化为了两个N / 2元素的DFT之和，以T(N)表示该算法的时间复杂度，我们有：



由主定理（Master Theorem）可知，复杂度为O(n\*lnn)。

根据此思路，我们很容易就能够实现一个递归形式的FFT。然而为了追求效率，实际中更为常见的是循环形式的FFT。循环形式，即从最底层的元素个数为 1的情况开始，每次利用公式④和公式⑤，在当前基础上获得元素个数乘2的结果，直到元素个数为N。在实现时，出于节省时间与空间，以及编码简单性的考虑，又会进一步使用进行码位倒读（在下一部分详述）等处理，方便进行蝶式算法，使得每次计算的结果仍能按当前顺序保存在结果数组中。

对于逆变换，我们有



由形式上我们可以看出，逆变换与正变换极为相似，只是多了一个系数1 / N，以及变为了。因而可以使用与上面相似的方法进行加速。

**二维傅里叶变换**

对于二维的傅里叶变换，我们有



注意到该公式还可以使用可分离的形式表示



通过这种形式，我们可以比较明显的看出，二维的DFT可以通过两个一维的DFT完成。对于每个x的取值，可以看做是一个一维的DFT变换F’(v)，即对于每个x，都可以使用O(NlnN)的时间计算出F’(v)对于v=1,...,N-1的取值。换言之，这是对于f(x,y)的一行（第x行）进行的傅里叶变换。以F(x,v)表示此步计算出的第x行信号在频率域的第v个分量，我们得到了另一个二维数组，原式转换为



很明显，这又是一个一维的傅里叶变换，只不过这次是沿着我们刚刚计算得出的二维数组的列的方向进行傅里叶变换。

由上面的分析，我们可以看出二维傅里叶变换可以由两次一维的一维傅里叶变换得到，每次一维变换都需要N\*O(NlogN) = O(N^2logN)的时间完成，总的时间复杂度仍为 O(N^2logN)。

**傅里叶变换的可视化**

在对傅里叶变换的结果进行可视化时，由于结果均为复数，无法直接转换为像素点，一般会将每个点上元素的模值与幅角分别展示。除此之外，还会对结果进行fftshift处理，从而将可视化图像中的亮点集中到图像中央，以便于结果的分析以及后续处理。

除此之外，由于傅里叶变换的结果中，元素与元素之间往往会有数量级的差距，导致在最终可视化中高亮度的元素会“掩盖”住低亮度的元素，我们往往会对结果取对数处理，之后线性归一化，从而缩小元素之间值的差距，提升视觉效果。

**源码解读**

作为后续所有操作的基础，一维傅里叶变换实现的效率直接影响整个系统的效率。算法的原理已在上一部分中叙述，主要思路是将N元素的FFT转换为两个N / 2元素的傅里叶变换加和。此部分比较关键的部分是码位倒读以及蝶式算法。

**码位倒读**

码位倒读是指二进制表示的数字首尾颠倒，重新按照十进制读数。进行此步操作可以实现即位计算，节省存储空间，方便编码。对于N位2进制数字，为实现码位倒读，我们首先需要将其二进制表示分为两半，对前、后半部分分别进行较小规模的逆转处理，之后再将前后部分交换位置。

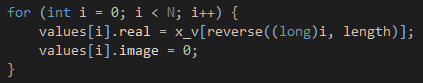
具体代码如下：



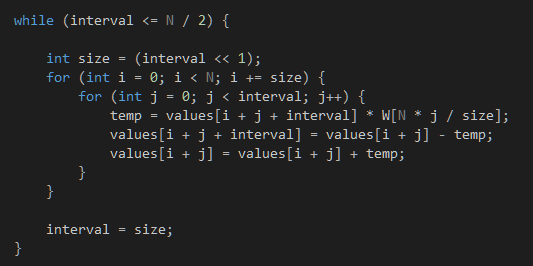
x表示待处理的数字，length表示其二进制表示的长度。使用half记录该二进制表示一半的位置，之后将lowwer赋为x二进制后半部分的值（可以使用与运算简单完成）。最后递归算出前后半部分逆转的结果，并将其位置交换。

**蝶式算法**

在完成码位倒读之后，便可以使用蝶式算法实现FFT的计算。具体实现如下：



values数组初始化为按照码位倒读顺序排列（原第i位的数字排列到i的码位倒读对应的位置上）的原始输入序列。

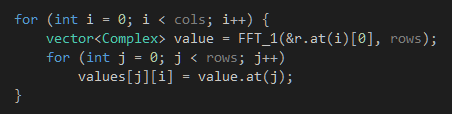


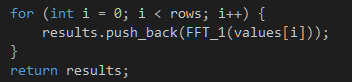
之后进入蝶式算法的主流程，每次迭代需要计算N / 2个蝶形，每个蝶形单元都有乘WkN的计算。

**二维傅里叶变换**

根据上一部分的分析，二维傅里叶变换可以由2N次一维傅里叶变换完成，首先需要延原二维信号列方向进行N次傅里叶变换，从而生成一个二维的中间结果数组，之后延中间结果数组的行方向再进行N次傅里叶变换，得到最终的变换结果。

为方便按列取出原始信号，需要原始二维信号进行转置处理，之后只需要取出每行数据进行一维傅里叶变换即可。具体代码如下：



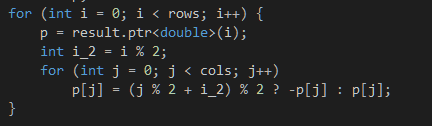
r为转置后的原始信号。注意此处由于是按列处理信号，结果仍将按列存放。之后按行对于中间结果进行傅里叶变换即可，具体代码如下：  
 

**结果的可视化**

在最终结果的可视化方面，将亮点集中到图像中心的方式有两种。一种是平移频率周期。具体做法是对原始信号进行预处理，对(x, y)处的信号进行乘以(-1)x+y处理。这种方式得到的结果与正常结果有如下的关系：

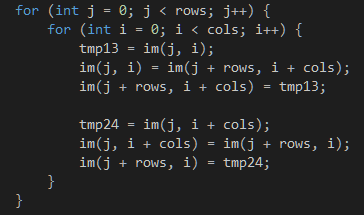


其中F为此方式获得的变换结果，F\_raw为原始方式获得的结果，信号矩阵的大小为M\*N。因而可以对周期进行平移，从而将(0, 0)点从边缘移动到中心。具体实现如下：

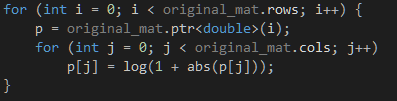


对于每个(i,j)位置的像素，只需判断(-1)i+j的符号即可。

除此之外，还可以在计算处频率结果后进行fftshift处理，大体思想是将结果均分成四块，然后将对角的结果两两交换位置，具体实现如下：



可视化部分同样十分重要的一环是对元素的normalize。首先需要逐元素取对数处理，



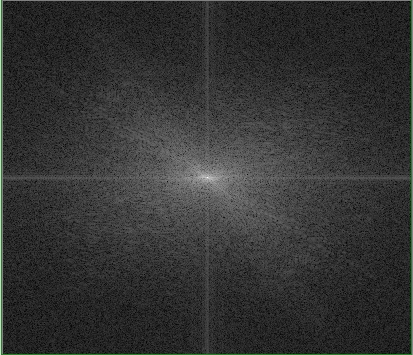
之后将每个元素的值都线性归一化到[0,1]即可。

**实现效果**

原始图像



DFT频谱



**离散余弦变换与逆变换**

**算法分析**

离散余弦变换也是数字图像处理中被广泛应用的技术。以C(k)表示对时间序列x(n)进行的一维离散余弦变换，其定义如下：





其逆变换定义如下：



显然，离散余弦变换（DCT）与DFT有紧密的联系，





根据此式可以看出，可以通过计算2N个元素的DFT（将原始x(n)延拓，将N到2N-1项延拓为0），来对N个元素的DCT进行求解。逆变换同理。

对于二维变换，我们有



同样注意到，此式与二维DFT具有类似的形式，因而可以采用类似的先行后列的方法进行计算。

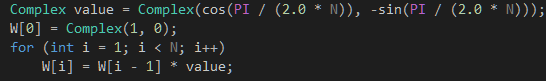
**源码解读**

根据上述分析，DCT部分的计算主要调用了FFT完成。然而，在调用之前，我们需要对信号进行延拓的预处理，

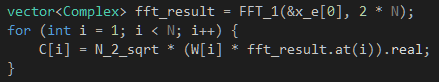


x为原始信号，x\_e为延拓后的信号，将后N个元素全部填充为0。

注意到DCT的定义式中fft的结果之前还需要乘以系数数组，该数组的具体计算如下



之后只需要进行fft运算，在每项前乘以上述的系数，在取出实部即可得到最终的结果



**实现效果**

原始图像



DCT频谱



**低/高通滤波器**

**算法分析**

借助傅里叶变换这一强大的工具，我们可以在频率域对图像进行分析预处理。一个比较直观的图像处理方法是对图像进行低/高通滤波，即从频率域中滤出频率较低/高的频率。一般来说，低通滤波能够保留图像大部分的特点，而忽略小部分细节；高通滤波，相应的则能突出图像的细节，达到锐化效果。

在进入具体滤波器的描述之前，还需要了解“振铃”的概念。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。

常见的低通滤波器主要有以下三种：理想低通滤波器（ILPF），巴特沃斯低通滤波器（BLPF），高斯低通滤波器（GLPF）。其中，ILPF能够对截止频率进行精确的控制，因而振铃现象最为严重；GLPF比较平缓，不会出现振铃现象；BLPF则介于两个极端之间，既能对截止频率做出较为准确的控制，又能抑制振铃现象。下面具体介绍三种滤波器的形式：

**ILPF**

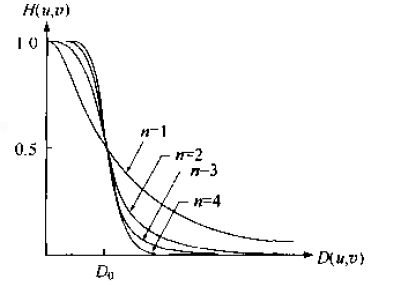


其中，D0为指定的非负数，D（u,v）是（u，v）点距频率矩形原点的距离。在经过了FFT节中所述中心化处理的频谱中，此距离即为该点到频谱中心的距离。从公式中可以看出，该滤波器在截止频率附近确实具有陡峭的变化，会从内部的1之间骤减到0。

**BLPF**



各个符号的定义同上。可以看出，当n较小时，滤波器的变化还是比较平缓的，这从下面的函数图像中可以较为明显的看出来



阶数从1到4的滤波器横截面\*

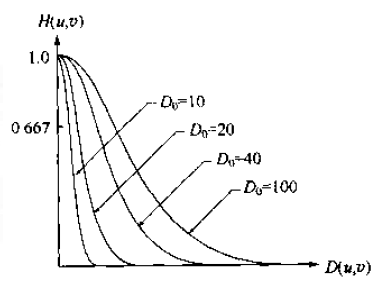
从上图我们还能看出，随着n的增长，滤波器的变化越来越陡峭，也即越来越接近ILPF。因而，在采用BLPF时，选取合理的n来在截止频率控制与抑制振铃间获得平衡十分重要。

**GLPF**



可以证明，高斯低通滤波器的傅里叶变换也是高斯函数，这意味着其不会出现振铃现象。

从函数图像中我们也可看出，截止频率前后，函数变化趋势的改变很小。



不同截止频率的GLPF横截面\*

在有了上述几个低通滤波器模型后，高通滤波器可以由 下面的关系式获得：



即高通滤波器可以通过所有被低通滤波器“挡住”的频率。据此，我们可以简单的获得理想高通滤波器（IHPF），巴特沃斯高通滤波器（BHPF），高斯高通滤波器（GHPF），具体形式如下：

**ILPF**



**BLPF**



**GLPF**



**高频提升过滤与高频加强滤波**

由于数字图像的低频保留了其主要信息，高频部分只涉及到图像中的细节。因而经过高通滤波器的频谱在转换到空间域后，背景的平均强度往往接近黑色，辨识度很差。因此，在实际中，我们往往会将一部分原始图像加到过滤后的结果中，即为**高频提升过滤。**具体来说，公式为



其中fhp（x，y）为高通过滤后的图像，a∈[0,+∞]，fhb高频提升过滤处理后的结果图像。该公式也可以转换为等价的频率域表示



其中Hhp为一高通滤波器，a为一正常数，Hhb为得到的高频提升滤波器。

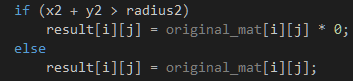
有时候，用一幅图像的高频成分强调增强的作用是有益的，因而我们可以在高通滤波器前乘以一个系数进行加强，即为**高频加强滤波器**，具体公式如下：



注意到当b=1时，高频加强滤波器退化到高频提升过滤，而当b>1时，高频得到加强。

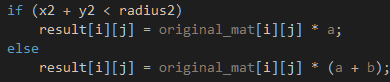
**源码解读**

**ILPF**



判断（i , j）点距离中心的距离是否大于规定的半径，如果大于则为高频，置为0，否则通过滤波器。

**IHPF**



此处实现了一个高频加强滤波器，偏移为a，增强系数为b。

**BLPF**



level表示滤波器的阶数，x2，y2分别代表（i，j）点与图像中心横、纵坐标插值的平方，radius2为规定半径的平方。

**BHPF**



此处实现了一个高频加强器，偏移为a，增强系数为b。

**GLPF**



变量的定义与之前相同，简单实现了GLPF公式

**GHPF**



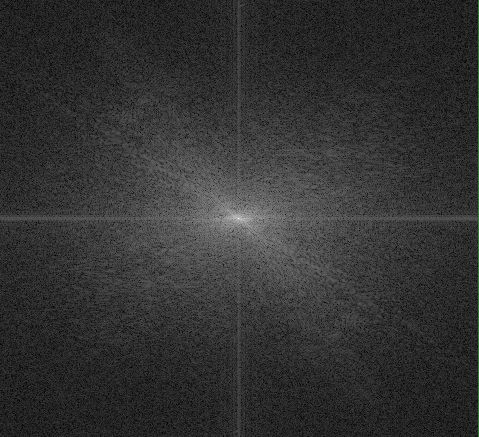
此处实现了一个高频加强器，偏移为a，增强系数为b。

**实现效果**

**原图**



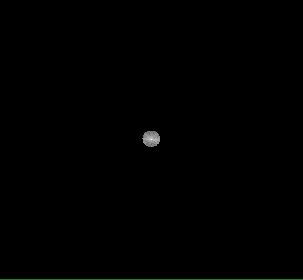
**原图频谱**



**ILPF（半径为15）**



**ILPF频谱**



低通滤波后的结果明显较为模糊，将图像钝化了。从频谱上来看，其只通过了内圈的频率。从图片中可以明显看到“振铃”现象。

**BLPF（半径为15，阶数为2）**



**BLPF频谱**

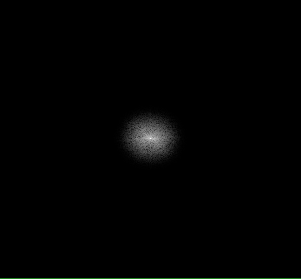


BLPF相较ILPF较为缓和，将结果比较，“振铃现象”也有所减弱。

**GLPF（半径为15）**



**GLPF频谱**

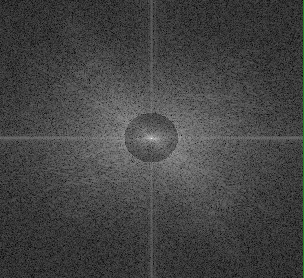


GLPF为三者中最为缓和的滤波器，其几乎没有“振铃”现象。同时从频谱中也可看出，其滤过的频率范围也是最大的。

**IHPF（半径为50，a=0.3，b=1.5）**



**IHPF滤波器**

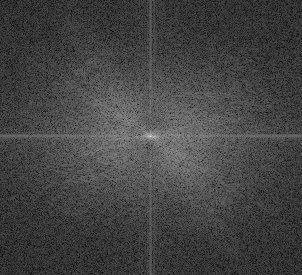


高通滤波器能够突出图片细节，对图片进行锐化处理。其突出了频谱中外围频率的强度，从频谱中可以明显看出内圈的截止频率。IHPF为三种滤波器中效果最为尖锐的，从图中可以看出，振铃现象极为严重。

**BHPF（半径为50，阶数为2，a=0.3，b=1.5）**



**BHPF频谱**

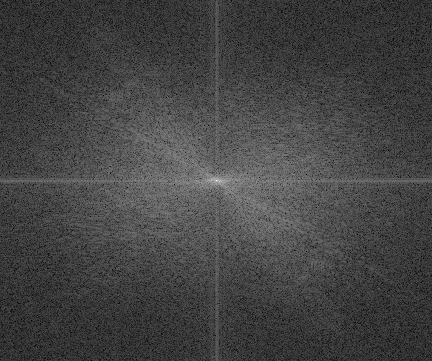


BHPF的效果介于两者之间。一定程度上缓解了振铃现象，同时从频谱中也能较为清晰的看出内圈的截止频率。

**GHPF（半径为50，a=0.3，b=1.5）**



**GHPF频谱**



GHPF的效果最为缓和，从频谱中几乎看不出截止频率，同时其在实现了锐化处理的同时，也抑制了振铃现象。