傅里叶变换在图像处理中的应用

姓名：邓栩瀛 学号：21307035

摘要：

傅里叶变换是一种将时域信号转换为频域信号的数学方法，可以用来分析和处理图像中的信息特征。本文主要介绍了傅里叶变换在图像处理中的基本原理和性质，以及它在图像增强[2]、图像去噪、图像分割、图像特征提取和图像压缩等方面的应用。傅里叶变换在图像处理中能够高效地进行图像存储和图像处理，使用新的方法和新的手段来处理图像以彰显傅里叶变换数学方法的价值所在。

1. 引言

图像处理是指对数字图像进行分析、处理和转换的技术，而这涉及到计算机视觉、人工智能、模式识别等多个领域，有着广泛的应用场景，例如医学影像、遥感监测、生物识别、视频压缩等等。

图像处理的目标是提高图像的质量，增强图像中的信息，或者从图像中提取有用的特征，从而对各个领域的研究提供强有力的帮助。

傅里叶变换是一种将时域信号转换为频域信号的数学方法，可以用来分析和处理图像中的信息特征。傅里叶变换可以将图像从空间域转为频率域，或者从频率域转换为空间域。在频率域中，能够更加容易地观察和处理图像中的低频分量和高频分量，从而实现对图像的增强[2]、去噪、分割、特征提取和压缩等操作。

2. 方法

2.1 傅里叶变换的基本原理

傅里叶变换是一种将时域信号转换为频域信号的数学方法，用来分析和处理信号中的信息特征。任何一个周期函数都可以表示为不同频率、幅度和相位的正弦函数或余弦函数之和，即傅里叶级数。对于非周期函数，但是满足一定条件（如绝对可积），也可以表示为不同频率、幅度和相位的正弦函数或余弦函数的积分。

对于一个离散函数 f[n]，其离散傅里叶变换 F(k)定义为：

$$F(k) = \sum_{n=0}^{N-1} f[n] e^{-j2\pi kn/N}$$

其中 k 表示离散频率索引，N 表示采样点数。

对于一个二维离散函数 f(m,n)，其二维离散傅里叶变换 F(p,q)定义为：

$$F(p,q) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi(pm/M+qn/N)}$$

2.2 傅里叶变换的性质

傅里叶变换具有一些重要的性质，用来分析和处理图像中的信息特征。以下是一些常用的性质：

线性性：傅里叶变换可以将线性组合的信号转换为频域中的线性组合。若 $x_1[n] \leftrightarrow X_1(e^{j\omega})$ ，$x_2[n] \leftrightarrow X_2(e^{j\omega})$ ，则有$a_1x_1[n] + a_2x_2[n] \leftrightarrow a_1X_1(e^{j\omega}) + a_2X_2(e^{j\omega})$。

共轭与共轭对称性：傅里叶变换可以互换时域和频域。若$x[n] \leftrightarrow X(e^{j\omega})$，则 $x^*[n] \leftrightarrow X^*(e^{-j\omega})$。

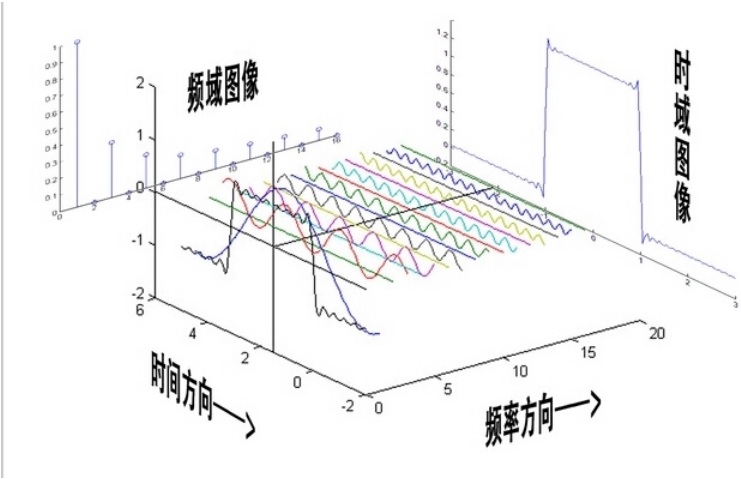
平移性：时域中的平移对应于频域中的相位移动，频域中的平移对应于时域中的调制。若$x[n] \leftrightarrow X(e^{j\omega})$，则$x[n - n_0] \leftrightarrow e^{-j\omega n_0} X(e^{j\omega})$；若$x[n] \leftrightarrow X(e^{j\omega})$，则 $e^{j\omega n_0} x[n] \leftrightarrow X(e^{j(\omega-\omega_0)})$。

卷积定理：时域中的卷积对应于频域中的乘法，时域中的乘法对应于频域中的卷积。如果$x_1[n] \leftrightarrow X_1(e^{j\omega})$ ，$x_2[n] \leftrightarrow X_2(e^{j\omega})$ ，则$x_1[n] * x_2[n] \leftrightarrow X_1(e^{j\omega})X_2(e^{j\omega})$。

时间反转：若$x[n] \leftrightarrow X(e^{j\omega})$，则$x[-n] \leftrightarrow X(e^{-j\omega})$。

2.3 图像傅里叶变换原理

傅里叶变换常用于数字信号处理，它的目的是将时间域上的信号转变为频率域上的信号。随着域的不同，对同一个事物的了解角度也随之改变，因此在时域中某些不好处理的地方，在频域就可以较为简单的处理。同时，可以从频域里发现一些原先不易察觉的特征。傅里叶定理指出“任何连续周期信号都可以表示成（或者无限逼近）一系列正弦信号的叠加。”



（图片来源：https://blog.csdn.net/weixin_30654419/article/details/ 97004919）

傅里叶变换可以应用于图像处理中，经过对图像进行变换得到其频谱图。从谱频图里频率高低来表征图像中灰度变化剧烈程度。图像中的边缘信号和噪声信号往往是高频信号，而图像变化频繁的图像轮廓及背景等信号往往是低频信号。这时可以有针对性的对图像进行相关操作，例如图像除噪、图像增强和锐化等。

二维图像的傅里叶变换数学公式如下：

$$F(k,l)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}f(i,j)e^{-i2\pi\left(\frac{ki}{N}+\frac{li}{N}\right)}$$

$$e^{ix}=\cos x+i\sin x$$

其中 f 是空间域值，F 是频域值。

3. 实验结果

3.1 使用 Numpy 实现傅里叶变换在图像处理中的应用[1]

Numpy 中的 FFT 包提供了函数 np.fft.fft2()，可以对信号进行快速的傅里叶变换[3]，输出结果是一个复数数组，函数原型为：

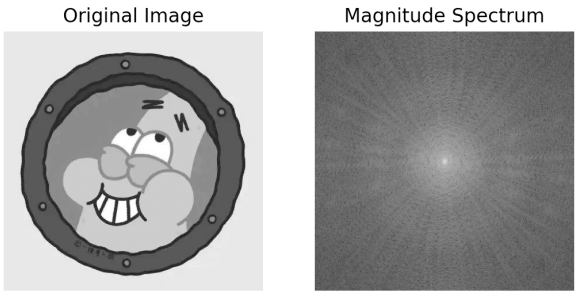
```
numpy.fft.fft2(x, s=None, axes=(-2, -1), norm=None)
```

其中，参数 x 表示输入的二维数组，参数 s 表示输出数组的形状；参数 axe 表示在哪些维度上进行傅里叶变换；参数 norm 表示归一化方式。

```
f = np.fft.fft2(img) # 快速傅里叶变换算法得到频率分布
# 默认结果中心点位置是在左上角,调用 fftshift()函数转移到中间位置
fshift = np.fft.fftshift(f)
```

```
#fft 结果是复数，绝对值结果是振幅
magnitude_spectrum = 20*np.log(np.abs(fshift))
```

运行结果如图：左边为原始图像，右边为频率分布图谱，越靠近中心位置频率越低，越亮（即灰度值越高）的位置代表该频率的信号振幅越大。



3.2 使用 Numpy 实现傅里叶逆变换在图像处理中的应用[1]

傅里叶变换的逆操作，就是将频谱图像转换为原始图像的过程。在前面的部分，我们通过傅里叶变换将原图转换为频谱图，并对高频（边界）和低频（细节）部分进行处理，现在通过傅里叶逆变换将频谱图恢复为原始效果图。频域上对图像的处理会反映在逆变换图像上，从而可以更好地对图像进行处理。

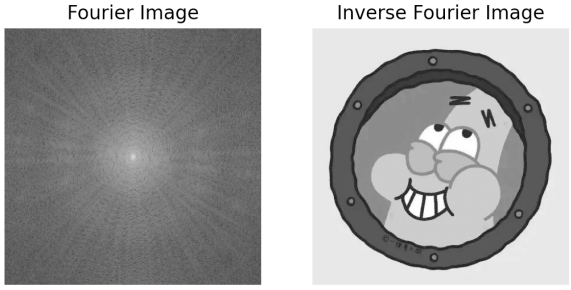
对二维数组进行离散傅里叶逆变换（IDFT）的函数原型如下：

```
numpy.fft.ifft2(a, s=None, axes=(-2, -1), norm=None)
```

其中，参数 a 表示输入的二维数组，参数 s 定义输出数组的形状，参数 axes 表示需要进行 IDFT 的轴，参数 norm 表示归一化方式，输出结果是一个与 a 形状相同的复数数组。

```
inverse_shift = np.fft.ifftshift(fshift)
inverse_img = np.fft.ifft2(inverse_shift)
inverse_img = np.abs(inverse_img)
```

运行结果如图：左边为频谱图，右边为傅里叶逆变换后得到的效果图，从而实现在频谱图上对图像进行处理的过程。

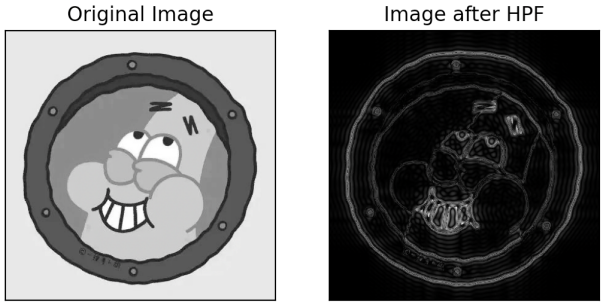


3.3 使用 Numpy 实现图像的高通滤波处理[1]

用一个 60*60 的矩形窗口做掩膜来去除低频部分，然后使用 np.fft.ifftshift() 函数进行逆平移操作，这样直流部分又回到左上角。最后用 np.ifft2() 函数进行 FFT 逆变换。

```
rows, cols = img.shape
crow,ccol = rows/2 , cols/2
# 掩膜，去除低频部分
fshift[crow-30:crow+30, ccol-30:ccol+30] = 0
f_ishift = np.fft.ifftshift(fshift) # 逆平移
img_back = np.fft.ifft2(f_ishift) # FFT 逆变换
img_back = np.abs(img_back) # 取绝对值
```

结果如图，左图为原始图像，右图为高通滤波后的图像。高通滤波实际上是一种边界检测。



4. 总结

使用 Numpy 实现傅里叶变换在图像处理的应用中，FFT 算法的实现是高效的，可以快速地计算出图像的傅里叶变换和逆变换，同时可以得到准确的傅里叶变换结果，加速图像处理的过程和提高图像处理的质量。

然而，在进行离散傅里叶变换时，如果信号的周期不是采样周期的整数倍，可能会出现频谱泄露等问题，即信号的高频部分会泄漏到低频部分，或者低频部分会泄漏到高频部分，从而影响到傅里叶变换结果的准确性。此外，如果输入图像的边缘没有进行合适的处理，可能会产生边缘效应，即可能会出现伪像或者振铃效应。

为解决如上问题，可以从以下几方面进行优化：其一，可以使用窗函数对输入信号进行处理，然后再进行傅里叶变换，可以在一定程度上缓解频谱泄露的问

题；其二，对输入信号进行零填充，以增加计算的频率分辨率，减少频谱泄露的问题；其三，使用零填充、边缘扩展等方法来处理输入图像的边缘，以减少边缘效应对实际处理的影响。

5. 参考文献

[1]OpenCV.*Fourier Transform - OpenCV Python Tutorials*[EB/OL]. 2018-08-17

[2] 王永会，陈荣.*基于分数阶傅里叶变换和频谱增强的路面裂缝图像识别方法*[J]. 计算机应用，2020

[3] 曹端良.*基于快速傅里叶变换的图像融合研究*[D].暨南大学，2019

6. 收获与感想

在《信号与系统》课程的学习中，我了解了信号与系统的基本概念，包括信号的分类、信号的时域和频域表示、系统的分类与特性等，学会了使用傅里叶变换和拉普拉斯变换对信号进行频域分析和系统的稳定性分析。

而在学习中，我深刻地理解了信号与系统在众多领域的研究中的重要性，是电子、通信、控制等学科的基础，对于进一步的研究有着重要的意义。同时，我还认识到信号处理的重要性及其广泛应用，包括音频、视频、图像等领域。而在这门课程中，通过编程实现信号的相关处理，我加深了对信号与系统理论的理解，同时也提高了自身的编程能力。