**Assignment7**

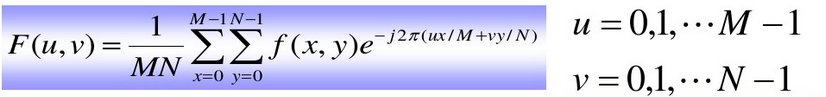
**Part 1 code**

详见FourierTransform.cpp。

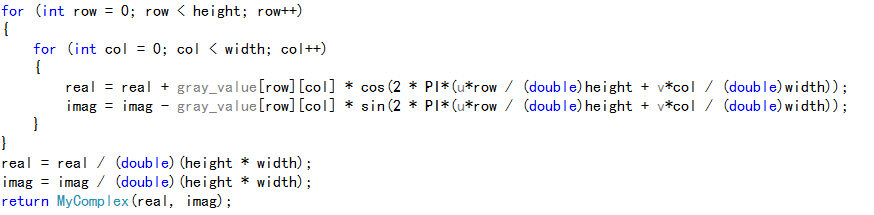
**Part 2 explanation**

1. **傅里叶变换**

对应的公式：

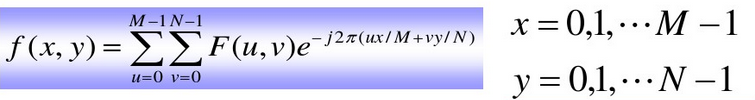


这里是代码中的实现：

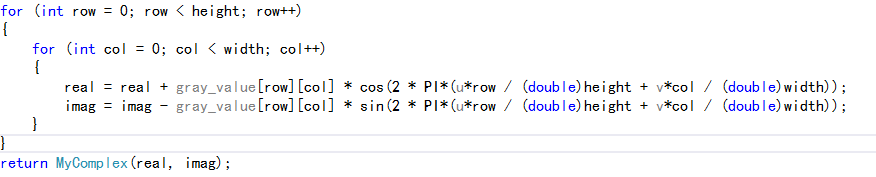


1. **逆傅里叶变换**

对应的公式：

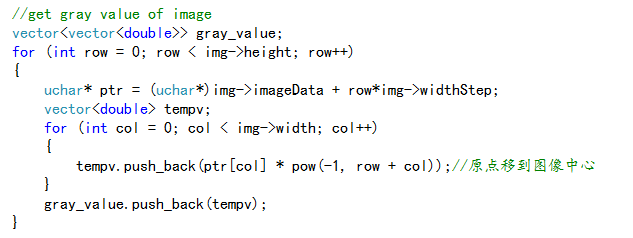


代码中的实现：



1. **载入原图**

在载入原图时，对灰度做了变换，使得最后的频谱图的原点在图片的中心。

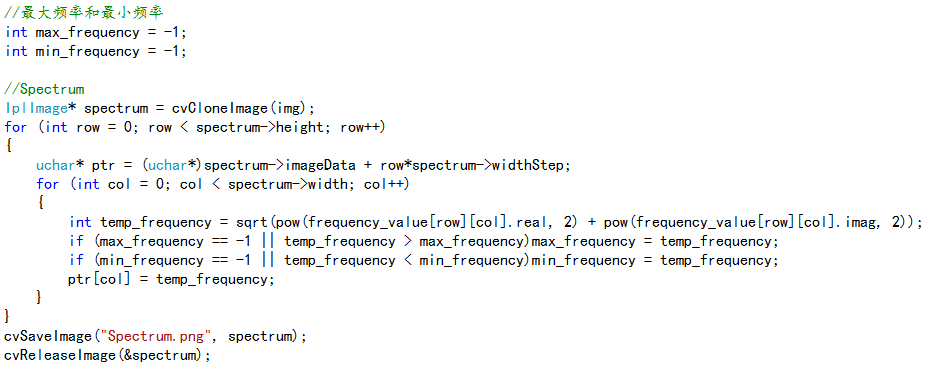


1. **计算频谱图：**

对应的公式：（这里R(u,v)是实部，I(u,v)是虚部）



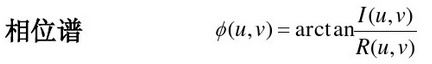
代码中的实现：



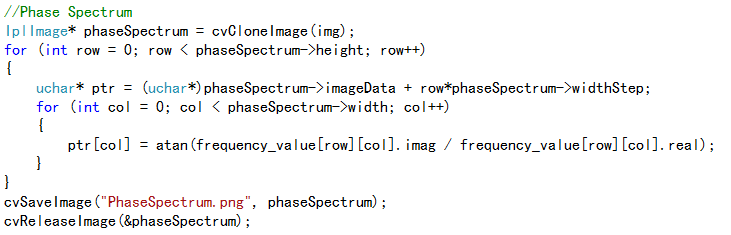
其中max\_frequency和min\_frequency是最大和最小频率。

1. **计算相位谱图**

对应的公式：

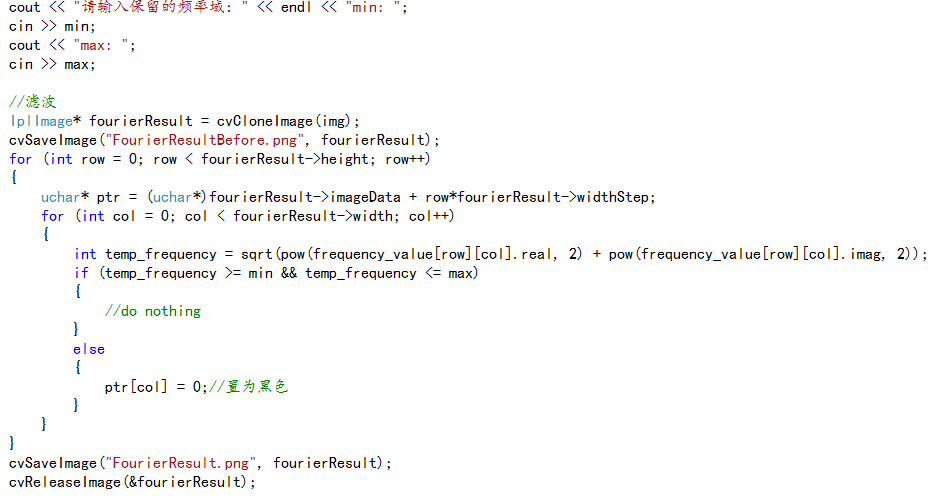


代码中的实现：



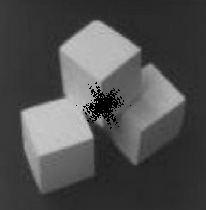
1. **滤波**

输入保留的频率域，之后进行滤波：

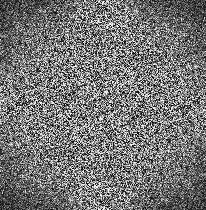
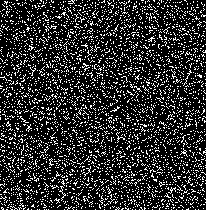


这里的max和min是输入的频率域。

**Part 3 output**

原图 滤波后

频谱图 相位谱图

**Part 4 problems**

虽然傅里叶变换是完全按照公式来计算，但是得到的频谱图跟网上的频谱图比起来差异较大，不知道是哪里有问题...

而且傅里叶变换在计算每个点时都要遍历整张图的信息，这样计算的次数就是N^2，N是图像中像素点的个数，导致运行时间较长。