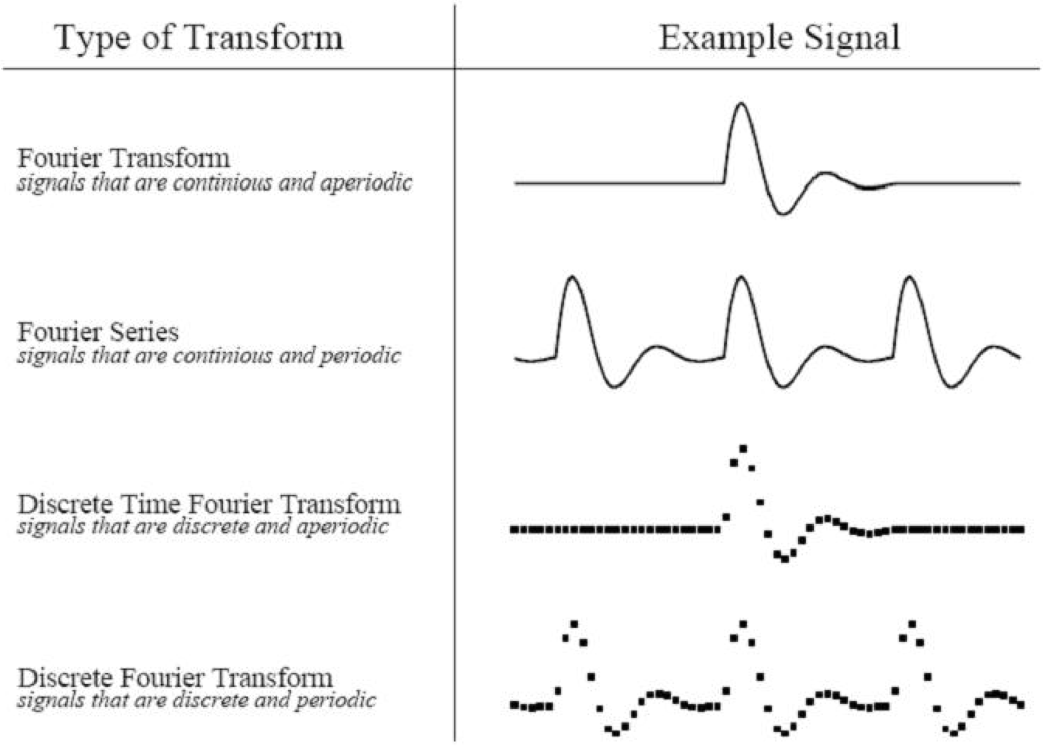
快速傅里叶变换–用于变调

曹逸君 1352872 张思雨 1352949

对傅里叶变换的理解

看了视频课、各种博文。对傅里叶变换(FFT)理解如下：
傅里叶变换有各种变体，
1. 非周期性连续信号 傅立叶变换（Fourier Transform）
2. 周期性连续信号 傅立叶级数(Fourier Series)
3. 非周期性离散信号 离散时域傅立叶变换（Discrete Time Fourier Transform）
4. 周期性离散信号 离散傅立叶变换(Discrete Fourier Transform)



实际信号处理中，都是用的离散傅立叶变换(DFT)，因为计算机能处理得只能是离散得值。

而快速傅里叶变换**(FFT)**则是将傅里叶变换的时间复杂度降至$n * log(n)$的改进算法。

傅里叶变换总的来说是将函数从基于时域和频域得相互转换得变换。
或者说成将任意(连续周期，数学上得要求，实际操作上可以对任意数据进行。)函数分解为多个正余弦函数。

傅里叶矩阵与傅里叶变换

复矩阵

1. 复向量的模 复向量 $z \in C^n$ 其模 $|z| = z^H z = \bar{z}^T z$
   zH表示对向量z的转置并共轭，H代表埃尔米特Hermite

2. 复向量的内积：$x^H y$

3. 埃尔米特矩阵：$A^H = A$ 当A是实矩阵时，即为对称矩阵。
4. 对称矩阵和埃尔米特矩阵的特征值是实数，特征向量相互垂直。
5. 复矩阵正交称为酉矩阵(unitary)，有$Q^H Q = I$ 即$Q^{-1} = Q^H$

傅里叶矩阵

$$F_n = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 & \cdots & w^{(n-1)} \\ 1 & w^2 & w^4 & w^6 & \cdots & w^{2(n-1)} \\ 1 & w^3 & w^6 & w^9 & \cdots & w^{3(n-1)} \\ 1 & \vdots & \vdots & \vdots & \ddots & w^{n(n-1)} \\ 1 & w^{(n-1)} & w^{2(n-1)} & w^{3(n-1)} & w^{n(n-1)} & w^{(n-1)^2} \end{pmatrix}$$

傅里叶矩阵是个特殊的酉矩阵。其中w满足$w^n = 1$。用复指数表示为$w = e^{\frac{i2\pi}{n}}$。PS:上帝公式$e^{i\pi=-1}$。

将原始数据向量左乘傅里叶矩阵就完成了傅里叶变换。
左乘傅里叶矩阵的逆矩阵$A^{-1} = A^H$就是傅里叶逆变换。

快速傅里叶变换(FFT)

由于傅里叶矩阵各列正交。又呈实矩阵那样得对称性。故可以将其分解。

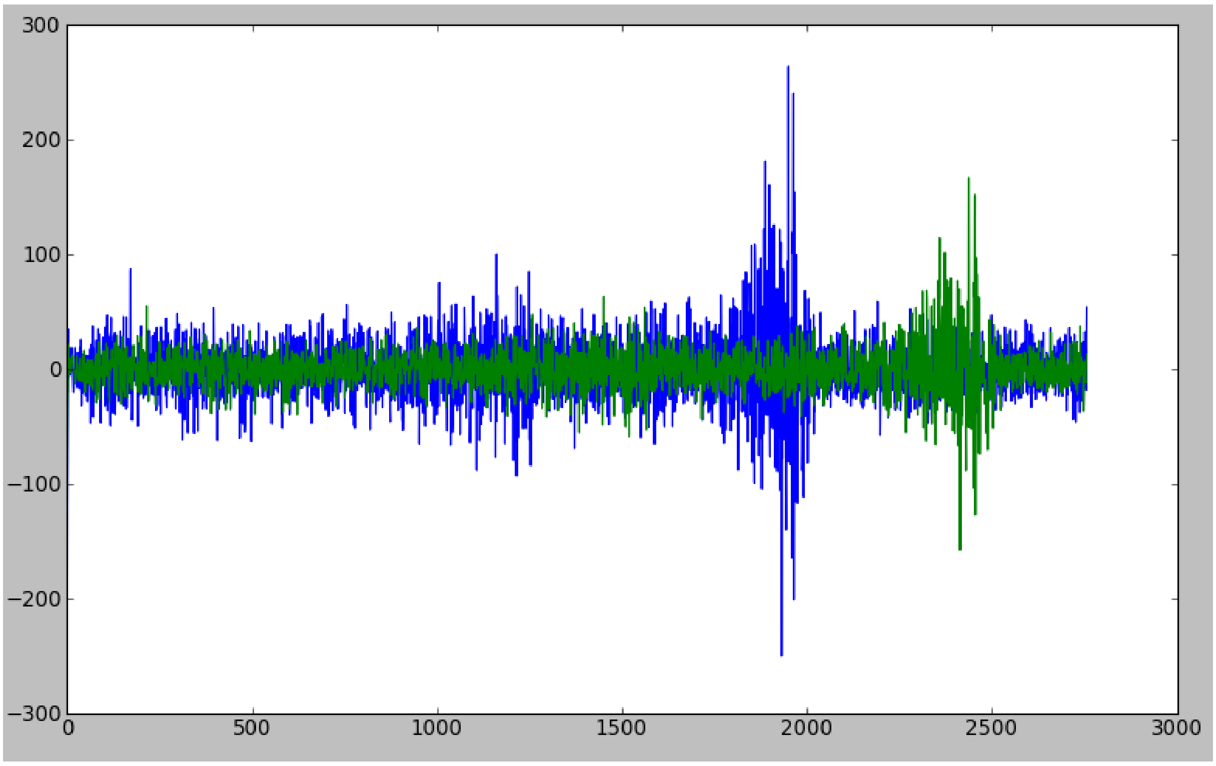
$$\left(F_{64} \right) = \begin{pmatrix} I & D \\ I & -D \end{pmatrix} \begin{pmatrix} F_{32} & 0 \\ 0 & F_{32} \end{pmatrix} \left(P \right)$$

P是奇偶置换矩阵。$D = \begin{pmatrix} 1 & & & & \\ & w & & & \\ & & w^2 & & \\ & & & \ddots & \\ & & & & w^{31} \end{pmatrix}$。并且$w_{32} = {w_{64}}^2$。

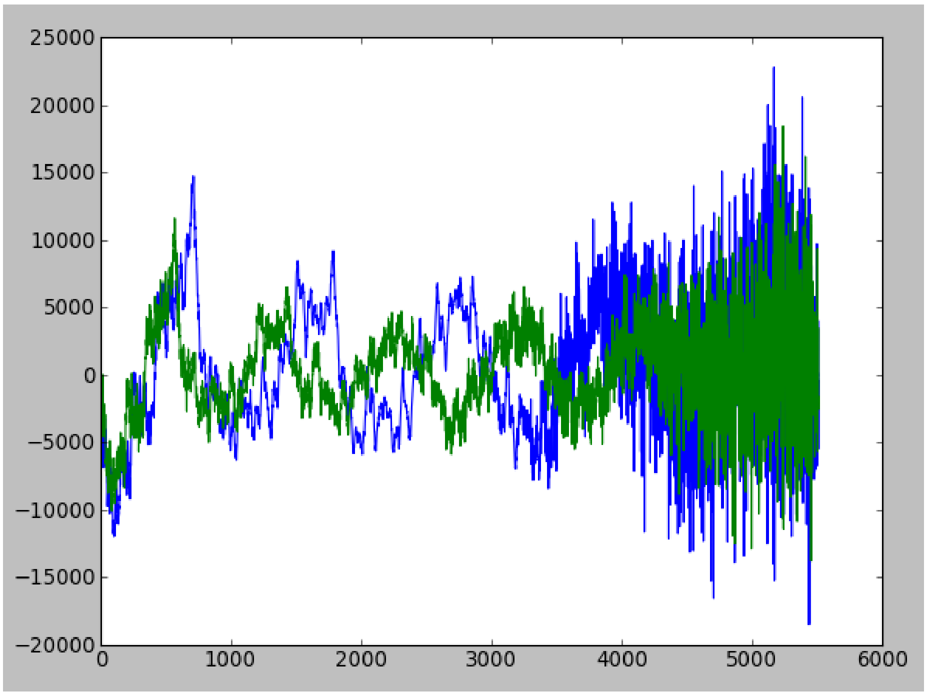
如上分解可以迭代下去到一阶傅里叶矩阵。而计算量也由n^2降为$\frac{1}{2}\, nlog(n)$。

快速傅里叶变换用于变调

过程如下：将音频采样(振幅对于时间得函数上得点)通过快速傅里叶变换转为不同频率声音的振幅分布。然后就可以方便的对不同频率进行操作了。比如这里我们将整个频率调高/调低(变得像女声/男声)。然后再通过逆傅里叶变换转换回振幅对于时间的分布数据。



图中蓝色为原始音频通过傅里叶变换得到的频谱。[横轴为频率，纵轴为振幅。]
绿色的是整体提升频率后得频谱。
绿色整体被拉长，表明频率整体升高。同时振幅整体缩小。[衰减2Db以降噪。]



上图蓝色为原始音频的波形图，[横轴为时间，纵轴为振幅。]
可以看到时间长度未变。频率却升高了。[波峰更加密集]

以上两图是1.25秒长度得小块数据，挑了两个比较清楚得。

源码和文件将和文档一起打包。

引用：
Numpy fft 手册
快速傅里叶变换将歌曲变调
介绍FFT的博文
MIT线性代数公开课，复数矩阵和快速傅里叶变换
线性代数导论27——复数矩阵和快速傅里叶变换,笔记

> Written with StackEdit.