音频识别课程报告

一.音频识别的任务描述

音频识别的工程可以按照时间顺序分为以下几个步骤:

1)识别准备

a.获得音频数据库

b.对原始音频进行预处理,去除多余细节,以一定的步长采样源文件.

2)识别主体

a.获取待识别音频片段

b.对待识别片段进行处理,处理至与数据库中的音频相同规格

c.将片段与数据库中的音频进行比对,统计相关性

d.将相关性最高的数据库中的对应音频作为可能的结果输出

3)识别结果记录

a.记录所生成的相关性图表

b.记录测试中识别错误的样本

二.信号相关的数学基础

信号的相关性定义为:

$$R(t) = \int f(x) \cdot g(x+t)\, dx$$

这里提供两种关于为何信号相关性是这样的解释.

对于离散的音频信号而言,不妨将其理解成一个带有顺序的实数集合,那么由柯西不等式的 n 阶推广可知,对于这个集合而言,必然存在着如下的关系

$$n_1 \in S$$

$$n_2 \in S$$

$$n \in S$$

其中 n1,n2,n 不重复取到 S 中

$$\sum n^2 \geq \sum n_1 n_2$$

将这个形式以类似柯西不等式的形式展开来就是:

$$a^2 + b^2 + c^2 + \cdots x^2 \geq ab + b^L + \cdots + xa$$

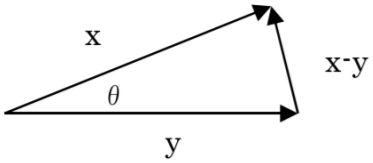
使用易于理解的语言描述,就是形如[a,b,c,⋯,x]的一组离散的信号,当自身对其自身的时候,每一个离散值之间的乘积大于等于不对齐时各项的乘积.所以随着 t 的变化,对信号的相关性的定义自然就成为了上文所示的样子.

还有一种理解方法,可以将 n 位信号理解成 n 阶向量,经由余弦定理可知

$$\|x-y\|^2 = E\left[(x(t)-y(t))^2\right] = E\left[x^2(t)\right] + E\left[y^2(t)\right] - 2E\left[x(t)y(t)\right]$$

$$\|x-y\|^2 = \|x\|^2 + \|y\|^2 - \frac{2E\left[x(t)y(t)\right]}{\|x\| \cdot \|y\|}\|x\|\|y\|$$

$$\cos\theta \Leftrightarrow \frac{E\left[x(t)y(t)\right]}{\|x\| \cdot \|y\|}$$



$$\phi_{xx}(\tau) = E[x(t)x(t+\tau)] = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)x(t+\tau)dt$$

即对于 t 位置而言,两个信号的相关性的大小与向量之间的余弦值，向量的模长有关。

求相关操作与卷积操作非常相似,所以可以利用卷积操作进行相关性的操作对于库中每一个音频文件做求相关性的操作,取其最高的相关性值作为文件相关性,比较各个文件的文件相关性值,最高的即为最具有相似性的结果.

三.具体项目中的几个要点和发现

1)在求解相关性时,直接使用经典定义公式中的方法进行相关性的求解效果远远差于使用 matlab 提供的 xcorr()方法的效果,前者在 1000 次随测试的结果为 36%,而后者大概为 96%.推测是 matlab 对于 xcorr()做了更多的操作,或者本项目中经典方法出现了错误。关于这一点，以及发现了问题所在，音频文件读取时默认列向量，而用作列向量反序的函数为 flipud（），并非课程 PPT 提示的行向量反序函数 fliplr（），造成了一定的问题

2)理论上,xcorr()处理时,会自动对短的信号进行补齐处理,再进行平移运算,所以可以利用这这个特点得到对应的音乐片段的具体时刻,这也正是 xcorr 返回值中的滞后指数.事实上，在后续的测试中，我发现 xcorr 的速度比经典的卷积方法显著地高，应该是使用了 FFT&IFFT 相关的方法加速了运算，在本项目中也模拟了这种运算方法，结果确实提高了运算速度。

3)利用 gpuArray 和并行处理的方式可以加速运算.