实时音乐和弦分析：一个使用Common Lisp Music音乐处理框架的系统

Takuya FUJISHIMA

音乐和声学计算机研究中心，斯坦福大学

斯坦福 加利福尼亚州 94305 美利坚合众国

[Fujishima@ccrma.stanford.edu](mailto:Fujishima@ccrma.stanford.edu)

摘要

本文介绍了一个从输入的声音信号中识别音乐和弦的实时软件系统。我设计了一个算法，并在CommonLispMusic/Realtime环境中实现了它。该系统运行在硅谷图形公司的工作站和英特尔PC平台上。实验证明，该系统甚至可以正确识别管弦乐队演奏的和弦。

关键词:音乐，声音，和弦，机器，识别，实时的，数字，人工智能

1 引言

音乐和弦识别是一个在识别特定的音高的同时，结合西方音乐中功能和声色彩的过程。传统上，识别单个音符被认为是一项对独立的音符进行识别的复调转录任务。这种方法在第一阶段受到识别错误的影响，这些错误来自于噪音，以及输入音频信号谱中音符的重叠谐波。这些不可避免的错误使和弦识别任务变得困难。

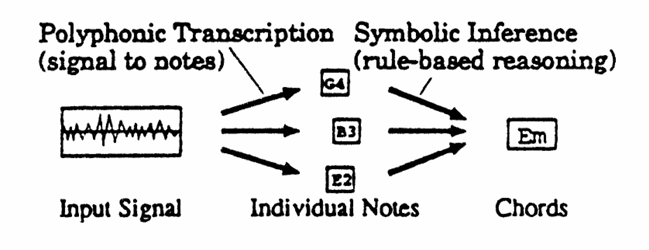


图1：传统方法

最近，莱曼提出了另一种方法。他的“简单听觉模型”(简称SAM)采用数值表示和信息处理的方法，避免了音符识别过程中的不可靠性，实现了一个鲁棒的识别系统。SAM的关键是使用12个半音音高类的强度图。该地图通过直接的数字处理从频谱中衍生出来，保存了更多的信息，实现了一个相比于音名系统更具有鲁棒性的系统。

根据SAM框架，我设计出了一个和弦识别算法，将在下文中提及

2算法

此和弦识别算法的概览如图2所示。

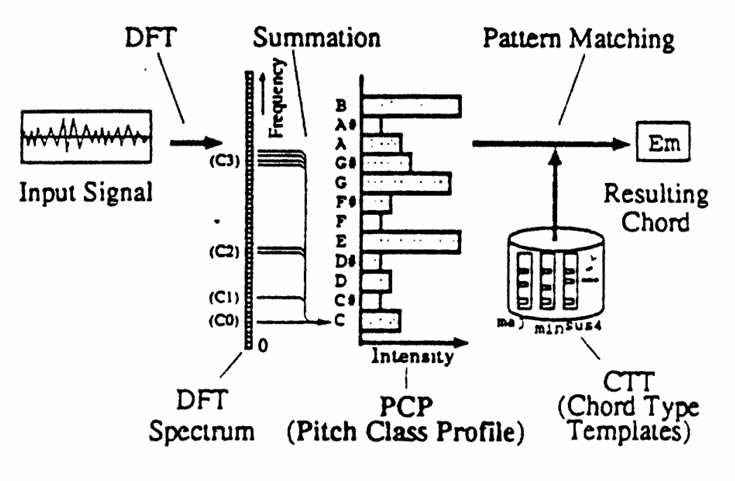


图2 和弦识别算法总览

这个算法首先将输入信号转换为一个离散傅里叶变换（DFT）频谱用于生成一个音级色谱图。然后进行模式匹配，来确定和弦类型以及根音。两个主要的优化方式健在后文详细介绍。

2.1 离散傅里叶变换频谱

首先，本算法将输入音频流进行分帧后转化为离散傅里叶变换频谱。设音频的采样率为，为长度为的音频帧，为数组下标。得离散傅里叶变换频谱在给定的下的结果为：

由于 为实数，即为整个频谱。

2.2 音级色谱图（PCP）

这个算法从中生成音级色谱图（PCP）。PCP类似于莱曼的SAM中的强度映射。它是一个十二维向量，表示十二个半音阶音符的强度。举例来说第一个PCP元素指示了C这个音符的总强度。

定义PCP向量

其中，为一个将频率点位映射至PCP下标的矩阵，其定于如下：

其中，为标准音频率，记为. 为频率点位的频率。注意，与可以在后续的使用时进行进一步的修改（详见2.4）.

2.3 模式匹配

对于获得的每个PCP，本算法使用和弦类型模版（CTT）进行模式匹配。PCP显示了一个和弦的信息。若一个以C为根音的类型为c的和弦具有音级p，那么=1，否则为0.举例来说，一个M7（大七和弦）的CTT为（1,0,0,0,1,0,0,1,0,0,0,1）。

我选择了27个和弦（表1），以是否在同一个八度中，分为两组：在同一个八度内（S1）与不同的八度内（S2）。

对比两种匹配方法：最临近算法与加权求和法。计算两种方法在所有和弦类型与偏置PCP矩阵的根音下的得分，选择最好的得分。

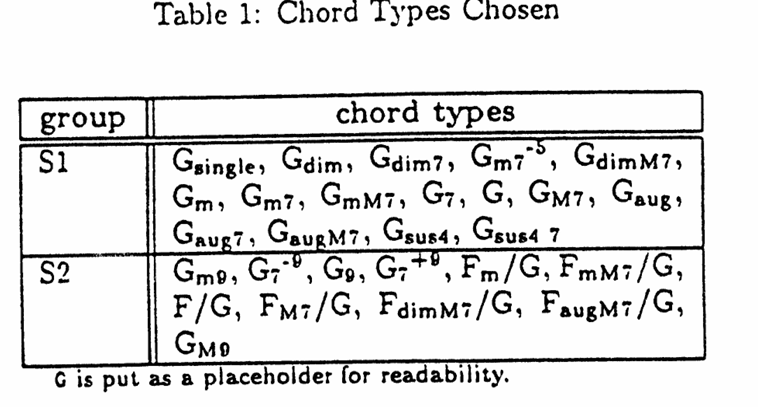


表1：选择的和弦类型

最临近算法：

用于选择最低得分进行匹配：

其中，TC(p)是一个典型的PCP轮转，从原始的CTTc(p)开始。

加权求和法：用于选择最大得分进行匹配：

其中Wc(p)是加权向量轮转，从CTTc(p)开始。因为在轮转时不同的和弦类型得到不同的权重，也体现了不同和弦类型中的音符数量与预测和弦类型的可能性。同时，负数权重在相似的和弦类型中能拥有更好的分离度。

2.4 启发

为了提高算法的整体性能，我介绍两种改进方法

1）PCP平滑处理：假设一个和弦通常持续几个PCP帧，这个算法可以将持续时间内的PCP向量使用平均数来进行平滑处理来减少噪声的影响。

2）和弦转换检测：和弦的转换通常是突然的，体现为PCP向量在十二维的空间中出现方向变化。因此，我们可以通过监控PCP向量的方向变化来识别和弦转换

介绍其他几个小的提升方法：

PCP预处理，包括了非线性缩放

消除在M(l)中不重要的区域

使用加窗函数

静音检测，防止非必要的分析

检测起音，来避免带有噪音的PCP

其中的细节不在此赘述

3 实现

CommonLispMusic/Realtime扩展(简称CLM/R)实现了该算法。CLM是一套灵活的基于common-lisp的综合与信号处理工具，由斯坦福大学CCRMA开发。CLM/r是CLM的实时扩展。它将LISP代码编译成二进制可执行文件，通过图形用户界面实现进实时的声音处理。该系统运行在硅谷图形公司02工作站(RS000/180mh z，Irix6.3)和英特尔PC平台(奔腾II/333mhz，RedHatLinux5.2)上

系统层级如图3所示。

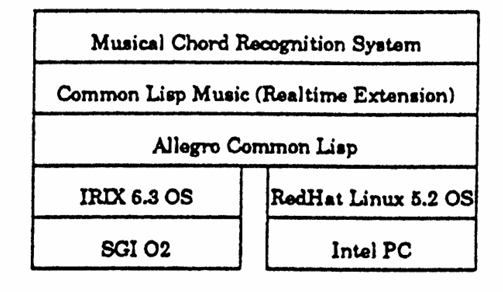


图3：系统层级

该系统在CLM/R平台上运行，可以接收来自线路输入或声音文件的音频信号，推断和弦，并在音频回放中实时显示结果。

4实验与实验数据

下述是我为本次实验选取的参数：采样频率fs=8.0kHz，单声道，DFT长度n=2048，无重叠，参考频率fref=65.4Hz。通过这些设置，系统产生了良好的PCP，而不会对来自YAMAHA SY-77音乐合成器的纯音输入产生主瓣干扰。

4.1 对键盘声音的响应

键盘声音的响应我用YAMAHApsr-520电子键盘的声音评估了这两种匹配方法的性能。线路输入是输入源。我选择了三种音色(Occalina，strings1和GrandPno)。在最近邻方法中，初始CTT在分离纯音Occalina，Stringsi和GrandPno中的所有27个和弦方面表现良好。在加权和方法中，我通过尝试和错误调整表，使其具有如下准确率:纯音和Occalina为100%(27/27)，GrandPno和strin g为93%(25/27)。

4.2对实际音乐的响应

我用从CD录音中提取的音乐声音测试了这个系统。为了可重复性，声音文件被用作输入源。系统对和弦类型和根音进行了一系列的猜测，我通过(#正确的猜测)/(#所有的猜测)给出了准确性。当和弦的所有音高类别都包含在实际和弦中时，和弦猜测就被认为是正确的。在下面的例子中，这个系统使用了从Smetana的“Moldau”(下图)的开头主题中摘录的一段50英寸的音频来演示。



输出样例

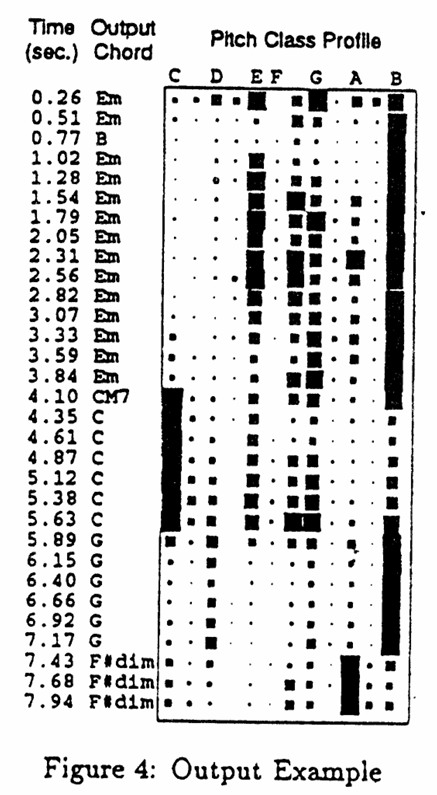


图4 输出样例

图4是结果的以加权求和法求出的PCP的前八秒。黑色方块的大小显示了PCP元素的值。

在4.10秒时，系统监听到了两个和弦：前一个Em和弦瑜伽下来的C和弦，输出了一个CM7和弦。

在7.43秒之后，系统输出的F#dim和弦，实际上是Am6或者F#dim7和弦。

示例这个例子是由最好的启发式参数(后面描述)和一个为实际音乐完全调整的表产生的。这是为了估计框架工作的潜在能力。最终的准确率达到94%(196/208)。通过实验，我发现为键盘输入而调整的表格在实际音乐分析中没有给出有意义的结果，反之亦然。

改进方法的作用

我测试了上述的改进方法，将启发参数的调整如下文所示：

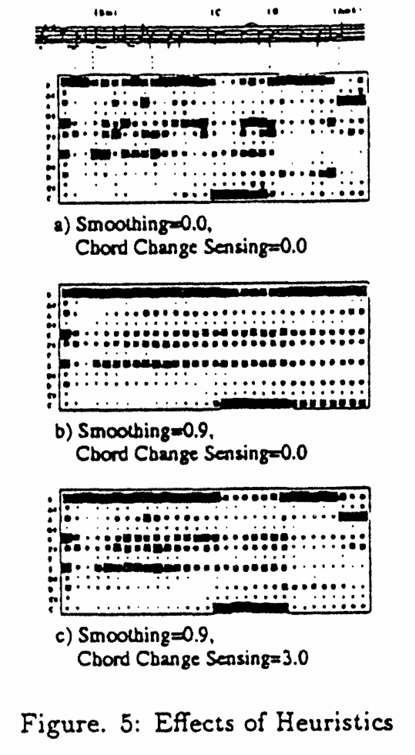


图5 改进方法的作用

图5展示了改进方法在PCP测算中的影响。时间轴自左向右。

1）未采用改进方法。PCP的噪音非常大。

2）PCP平滑算法减少了噪音，但是过度的平滑模糊了和弦转换点。

3）和弦转换测算增强了和弦转换点的测算。

图6是在不同的启发参数下的准确率：

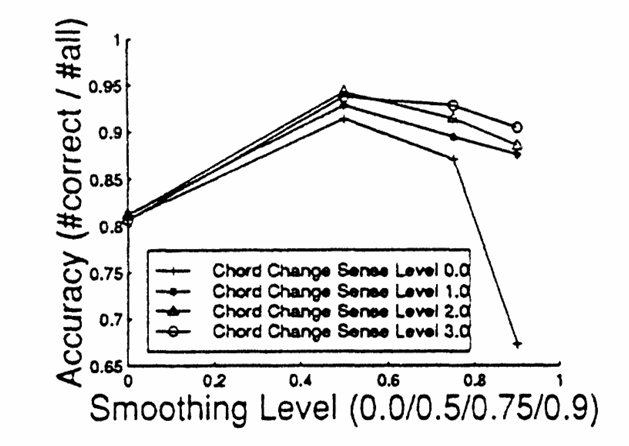


图6 准确率vs 启发参数

图6显示了在平滑等级0.5，和弦转换检测灵敏度为2.0时获得了最准确的结果。这也是在图4中用于举例的参数。

5 讨论

实验结果表明，该系统能够在信号中成功地识别音乐和弦。即使在复杂的多音域情况下，比如管弦乐，它也能指定正确的和弦名称。考虑到系统的规模(LISP源代码中有1371行)，这个系统的性能值得肯定。PCP被证明是一个合理的内部表示，它传达的信息足以确定和弦类型和根音，虽然它丢弃了低音信息。在PCP中引起“混淆”的和弦(表1中的S2)也可以被正确识别。

无论是加权和法还是最近邻法都能正确分离单音或复音的93%到100%的和弦。在实际音乐中，加权和法识别和弦的准确率高达94%，而最近邻法的准确率不高。因此，这些分类器至少在理想的输入情况下是足够的，但是更好的分类器可能对处理实际的音乐输入是必要的。两种保持边缘平滑的启发式算法将整体准确度提高了约10 %。未来的工作将包括一个更好的分类器，表调整方法，低音音符跟踪，使用音乐上下文的符号级别错误纠正。

6 致谢

本实验在斯坦福大学音乐和声学计算机中心的Jonathan Berger教授的指导下完成。我对它的支持与建设性的建议表示真诚的感谢。同时，感谢雅马哈集团的品牌主管Masatada Wachi先生，给予我开展此研究的机会。

参考文献

