



(12)发明专利

(10)授权公告号 CN 105227763 B

(45)授权公告日 2018.03.20

(21)申请号 201510549631.3

G10L 21/0272(2013.01)

(22)申请日 2015.08.31

(56)对比文件

(65)同一申请的已公布的文献号
申请公布号 CN 105227763 A

CN 101299344 A, 2008.11.05,
CN 101494049 A, 2009.07.29,
CN 104347096 A, 2015.02.11,
US 2012/0297959 A1, 2012.11.29,
WO 2009/031979 A1, 2009.03.12,
王志明等.《基于分层熵检测的音频分割算法》.《科学技术与工程》.2009,第9卷(第17期),第5012-5016页.

(43)申请公布日 2016.01.06

(73)专利权人 武汉工程大学
地址 430074 湖北省武汉市洪山区雄楚大街693号

梁伟等.《一种快速的音频分割与分类技术》.《第八届全国人机语音通讯学术会议论文集》.2005,第80-84页.

(72)发明人 冷娇娇 赵彤洲 方晖 李翔
李碧 翟畅

(74)专利代理机构 湖北武汉永嘉专利代理有限公司 42102

审查员 魏亚南

代理人 唐万荣

(51)Int. Cl.

H04M 1/725(2006.01)

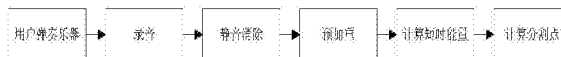
权利要求书1页 说明书4页 附图2页

(54)发明名称

一种在智能移动设备上实现的乐器音频实时分割方法

(57)摘要

本发明公开了一种在智能移动设备上实现的乐器音频实时分割方法,该方法能够在移动设备上实现了录音和分割的同步操作。为了达到实时的效果,本发明在进行了快速预处理以后,通过计算定长窗口内信号的短时能量以及寻找能量均值的突变来确定真实跳变点的存在,使得边采集边处理成为可能。对于乐器发声的音频信号,本发明可以在用户弹奏乐器的同时,对采集到的音频信号进行分割,并快速显示,为用户进行音频识别、获取语义结构化信息、进行音频检索等深层处理做准备。



1. 一种在智能移动设备上实现的乐器音频实时分割方法,其特征在于,包括以下步骤:

1) 采集音频数据:由移动通讯设备上的录音设备实时获取乐器的音频数据;

2) 前端静音消除:消除采集的音频数据中开始部分的静音数据;所述静音数据为原始音频数据中前端振幅小于 10^{-4} 的数据;

3) 对消除静音数据后的音频数据进行实时处理,具体包括:

3.1) 预加重数字滤波:对经过前端静音消除处理的音频信号进行预加重处理和数字滤波;

3.2) 分帧:采用交替分帧方法对连续的分帧进行平滑过渡;

3.3) 计算短时能量:对分帧后的每一帧信号,计算短时能量: $E = \sum_{m=0}^{n-1} x^2(m)$,其中参数n

为窗口长度,x(m)为信号序列;

3.4) 背景音消除:若帧信号能量值低于音频信号最大能量的0.1倍时,该帧信号为背景音,将背景音删除;

3.5) 查询分割点:对每相邻3帧的能量均值H进行比较,如果存在较大差距,则认为存在突变点,突变点为分割点,即满足: $H_1 > H_2 * Th$,其中,Th为判断前后相邻3帧的阈值系数;

4) 根据分割点所在的位置实现的乐器音频实时分割。

2. 根据权利要求1所述的乐器音频实时分割方法,其特征在于,所述步骤3.1)中,按照6dB/oct的比例对样本进行预加重处理;数字滤波采用以下公式: $y(n) = x(n) - a * x(n-1)$,其中参数a取值为0.9~1之间,x(n)为滤波前的采样信号,y(n)为滤波后的信号。

3. 根据权利要求1所述的乐器音频实时分割方法,其特征在于,所述步骤3.2)中,分帧策略如下:设定帧长 $L = m * 256$,其中m为1~16中的任意整数,帧移步长取帧长L的0.3~0.7倍。

4. 根据权利要求1所述的乐器音频实时分割方法,其特征在于,所述步骤3.5)中,Th取值范围为3~7之间。

一种在智能移动设备上实现的乐器音频实时分割方法

技术领域

[0001] 本发明涉及,尤其涉及一种在智能移动设备上实现的乐器音频实时分割方法。

背景技术

[0002] 音频分割是进行音频识别、获取语义结构化信息、进行音频检索的必要步骤。传统的音频分割方法是在获取全部的音频信息后才开始对音频进行分割,即没有实现录音及分割的同步操作。传统音频分割方法是在PC端进行计算,因PC机不具备便携性,限制了在智能移动通讯设备上某些需要进行音频分割应用的使用和推广。

发明内容

[0003] 本发明要解决的技术问题在于针对现有技术中的缺陷,提供一种在智能移动设备上实现的乐器音频实时分割方法。

[0004] 本发明解决其技术问题所采用的技术方案是:一种在智能移动设备上实现的乐器音频实时分割方法,包括以下步骤:

[0005] 1) 采集音频数据:由移动通讯设备上的录音设备实时获取乐器的音频数据;

[0006] 2) 前端静音消除:消除采集的音频数据中开始部分的静音数据;

[0007] 3) 对消除静音数据后的音频数据进行实时处理,具体包括:

[0008] 3.1) 预加重数字滤波:对经过前端静音消除处理的音频信号进行预加重处理和数字滤波;

[0009] 3.2) 分帧:采用交替分帧方法对连续的分帧进行平滑过渡;假设一帧的长度wlen,步长为inc,那么相邻两帧的重复信号长度即为wlen-inc;

[0010] 3.3) 计算短时能量:对分帧后的每一帧信号,计算短时能量: $E = \sum_{m=0}^{n-1} x^2(m)$,其中参

数n为窗口长度,x(m)为信号序列。

[0011] 3.4) 背景音消除:若帧信号能量值低于音频信号最大能量的0.1倍时,该帧信号为背景音,将背景音删除;

[0012] 其中,音频信号最大能量当时的音频信号最大能量,且音频信号最大能量为实时更新;

[0013] 3.5) 查询分割点:对每相邻3帧的能量均值 \bar{H} 进行比较,如果存在较大差距,则认为存在突变点,突变点为分割点,即满足: $\bar{H}_1 > \bar{H}_2 * Th$,其中,Th为判断前后相邻3帧的阈值系数;

[0014] 4) 根据分割点所在的位置实现的乐器音频实时分割。

[0015] 按上述方案,所述步骤2)中设原始音频信号前端振幅小于 10^{-4} 的信号是静音

[0016] 按上述方案,所述步骤3.1)中,按照6dB/oct的比例对样本进行预加重处理;数字滤波采用以下公式: $y(n) = x(n) - a * x(n-1)$,其中参数a取值为0.9~1之间,x(n)为滤波前的

采样信号, $y(n)$ 为滤波后的信号。

[0017] 按上述方案,所述步骤3.2)中,分帧策略如下:设定帧长 $L=m \times 256$,其中 m 为1~16中的任意整数,帧移步长取帧长 L 的0.3~0.7倍。

[0018] 按上述方案,所述步骤3.5)中, Th 取值范围为3~7之间。

[0019] 本发明产生的有益效果是:本发明方法针对移动端的乐器音频实时分割,适合移动端处理,能达到速度快且消耗资源少的优点,能实现对音频源进行边录音边分割操作,非常适用于对音频实时分割类应用。

附图说明

[0020] 下面将结合附图及实施例对本发明作进一步说明,附图中:

[0021] 图1是本发明实施例的方法流程图;

[0022] 图2是本发明实施例中计算短时能量的流程图;

[0023] 图3是本发明实施例中查询分割点的流程图。

具体实施方式

[0024] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅用以解释本发明,并不用于限定本发明。

[0025] 如图1所示,一种在智能移动设备上实现的乐器音频实时分割方法,包括以下步骤:

[0026] 1) 采集音频数据:由移动通讯设备上的录音设备实时获取乐器弹奏的音频数据;采样频率为44.1KHz或者22.1KHz,音频信号文件格式为mp3。

[0027] 2) 前端静音消除:消除采集的音频数据中开始部分的静音数据;在采样过程中,由于录音准备需要一段时间,导致采样数据的前端有若干时间内没有音乐发声,即弹奏的准备阶段,这段采集的振幅近似于0的背景音就是静音数据。显然应该消除这部分无声数据,以便去除冗余为能提高计算速度。本系统认为原始音频信号前端振幅小于 10^{-4} 的信号是静音,在本发明所需的用户环境下,这样的静音数据可以认为是完全无效的干扰数据,应全部予以删除。

[0028] 3) 对消除静音数据后的音频数据进行实时处理,具体包括:

[0029] 3.1) 预加重数字滤波:对经过前端静音消除处理的音频信号进行数字滤波;由于音频信号在传输过程中会有衰减,因此按照6dB/oct的比例对样本进行预加重处理,使得输出电平相不会产生较大偏差,同时还可以消除低频干扰。公式为: $y(n) = x(n) - a * x(n-1)$,参数 a 取值0.9~1之间的数。

[0030] 3.2) 分帧:音频样本属性在较短的时间内变化较为平缓,在此平缓变化的时间内所提取的音频特征保持基本稳定。因此,本发明采用交替分帧方法对连续的分帧进行平滑过渡,以保持其连续性。设定帧长 $L=m \times 256$,其中 m 为1~16中的任意整数,帧移步长取帧长 L 的0.3~0.7倍,即每相邻的两帧都存在70%~30%的重复数据,以此可以减少分割点的漏检率和误检率。

[0031] 3.3) 计算短时能量:对分帧后的每一帧信号,计算短时能量: $E = \sum_{m=0}^{n-1} x^2(m)$,参数n为窗口长度。

[0032] 3.4) 背景音消除:背景音的能量较弹奏音乐小,在稳定环境下,可以认为当能量值低于音频信号最大能量的0.1倍时为背景音,将此部分背景音予以删除,以便提高计算速度。

[0033] 3.5) 查询分割点:对每相邻3帧的能量均值 \bar{H} 进行比较,如果存在较大差距,则认为存在突变点(分割点),即满足: $\bar{H}_1 > \bar{H}_2 * Th$,其中,Th为判断前后相邻3帧的阈值系数,一般取3~7之间。

[0034] 4) 根据分割点所在的位置实现的乐器音频实时分割。

[0035] 本发明可采用Java语言进行具体程序设计,主要分为录音和分割两个模块。为了保证在移动通讯设备上能完成音频信号的实时处理,设计过程中开启两个后台线程,分别用于录音和分割。

[0036] 1) 用InitNotifications()函数作为初始化录音的通知线程,当录音的缓冲区满后把数据存入allData对象中,为后续音频数据的预处理、分割以及最终识别存储数据。

[0037] 2) 建立delSilence()函数,用以对前端静音数据的删除,即删除原始音频前端振幅小于 10^{-4} 的信号。

[0038] 3) 建立filter(short[] data,int z)函数,用以对静音删除后的信号滤波。滤波函数为filter(),参数取值0.97或0.98,此滤波函数与静音删除后的信号做卷积运算,结果存储于preProcess()函数中。

[0039] 4) 弹奏乐器过程中,一个音符被按下后,音频的振幅是逐渐衰减的。为防止高频的频段跌落,将音频信号进行交替分帧处理,处理函数为frame(int m,double c),其中m为帧长的截取系数,c为帧移系数,通常帧移系数为0.3~0.7之间,分帧结果存储在函数frameSignal()函数中。

[0040] 5) 对分帧后的结果frameSignal()计算短时能量,短时能量即为frameSignal()

中的每帧元素的平方和,即 $E = \sum_{m=0}^{n-1} x^2(m)$,结果由函数shortEnergy()传递。具体流程如图2所示。

[0041] 6) 为了加快查找分割点的计算速度,在进入分割点查找前需要删除离群点。经过计算短时能量后,对某些离群点产生的噪声会产生放大作用,因此,需要剔除这些离群点。与弹奏的音乐相比,噪声的能量要远远小于弹奏音乐,因此,本方法经过大量实验,认为当能量值低于音频信号最大能量的0.1倍时为噪声,删除这部分噪声。

[0042] 7) 经过平滑和去噪后,音频变化趋势非常明显。本方法通过判断每相邻3帧的平均值是否存在较大变化作为分割点的认定依据,认为如果相邻帧的变化非常明显,则认为有一个新的音符出现。此过程用函数findSegment(double[] shortEnergy,int BgTh)实现,其中BgTh为背景音的阈值系数,通常取0.1。如果某一帧的能量数据小于最大能量的BgTh倍,则认为该帧的不存在分割点,无需进行分割点查询。符合分割点查询的结果即为分割

点,用pos[]标记出现的位置。

[0043] 具体流程如图3所示。

[0044] 8) 根据分割点所在的位置即可快速实现的乐器音频实时分割,pos[]集合就是分割点所在的位置集合。

[0045] 应当理解的是,对本领域普通技术人员来说,可以根据上述说明加以改进或变换,而所有这些改进和变换都应属于本发明所附权利要求的保护范围。

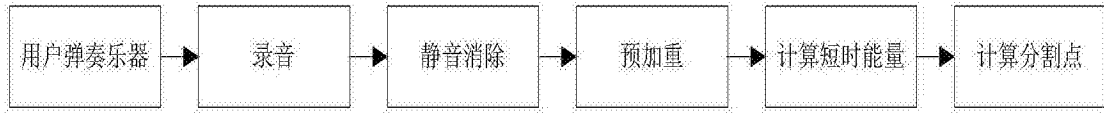


图1

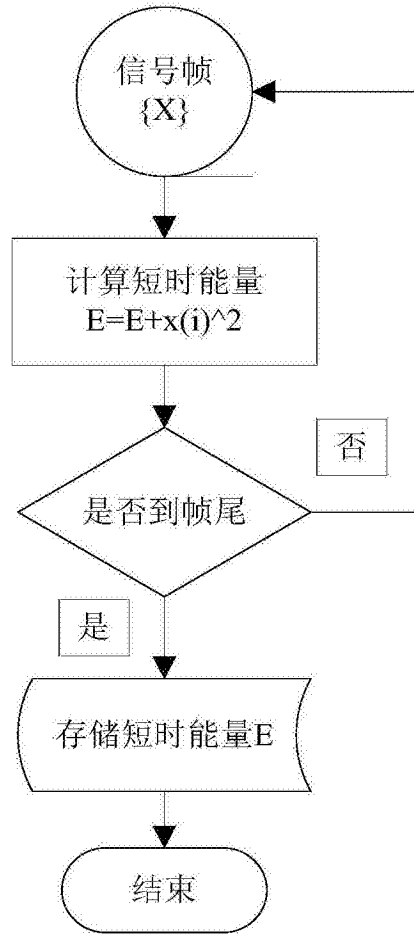


图2

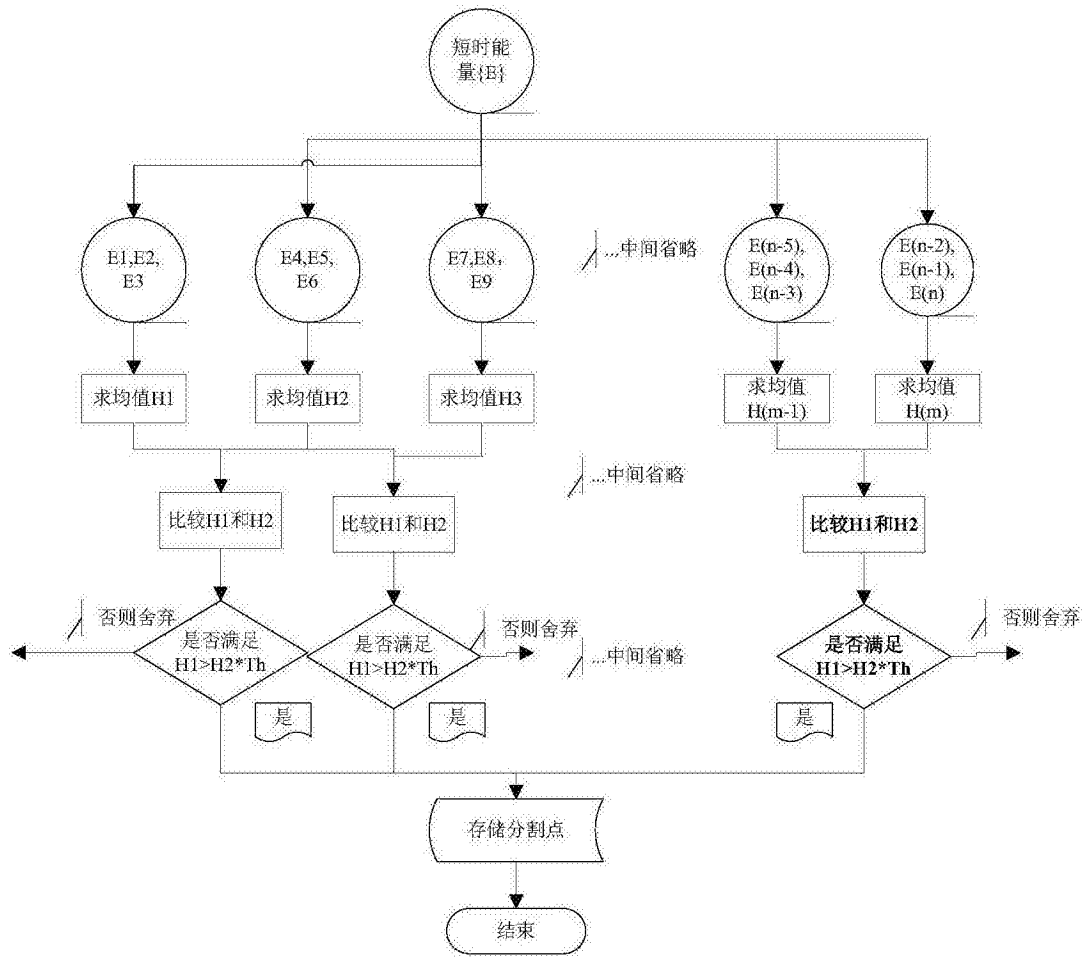


图3