

乐曲伴奏的生成方法、装置、设备、存储介质及程序产品

本申请要求于 2023 年 12 月 27 日提交的申请号为 202311824499.3、发明名称为“乐曲伴奏的生成方法、装置、设备、存储介质及程序产品”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请实施例涉及计算机技术领域，特别涉及一种乐曲伴奏的生成方法、装置、设备、存储介质及程序产品。

背景技术

艺术创作一直以来被认为是艺术家的专属领域，然而近几年来随着计算机技术的发展，艺术创作逐渐打破了传统壁垒，借助计算机技术可以人工制作音乐作品，使得音乐类型以及音乐内容拥有了更大的提升空间。

相关技术中，若希望基于音乐生成对应的音乐伴奏，通常采用基于深度学习的方法进行伴奏生成，借助样本音乐与对应的样本伴奏对模型进行训练，以使得模型学习到生成伴奏的方法；之后将需要生成伴奏的音乐输入训练后的模型，得到音乐伴奏。

上述过程过分依赖于样本音乐与样本伴奏之间准确的对应关系，该种方式虽然能够在一定程度上得到良好的音乐伴奏，但是忽略了音乐本身具有的乐理信息，容易导致音乐伴奏的乐曲风格与音乐本身存在较大差距，影响音乐伴奏在音乐领域的准确性。

发明内容

本申请实施例提供了一种乐曲伴奏的生成方法、装置、设备、存储介质及程序产品，能够通过对乐曲节拍以及乐曲音符进行量化处理，在乐曲旋律的限定将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现，提高乐曲伴奏的生成稳定性以及生成效果。所述技术方案如下。

一方面，提供了一种乐曲伴奏的生成方法，所述方法由计算机设备执行，所述方法包括：获取乐曲数据，所述乐曲数据被划分为至少两个乐曲节拍，所述乐曲节拍包括乐曲音符；

从所述乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息，所述节拍数据信息用于描述所述至少两个乐曲节拍的变化速度，所述和弦数据信息用于描述以所述乐曲节拍为单位提取的和弦单元，所述旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况；

根据所述旋律数据信息、所述节拍数据信息以及所述和弦数据信息，生成乐曲伴奏数据，所述旋律数据信息用于以所述乐曲数据的乐曲旋律限定所述乐曲伴奏的伴奏旋律，所述乐曲伴奏数据用于描述所述乐曲数据的伴奏情况；

基于乐曲伴奏数据执行音频数据渲染，得到所述乐曲数据对应的所述乐曲伴奏。

另一方面，提供了一种乐曲伴奏的生成装置，所述装置包括：

数据获取模块，用于获取乐曲数据，所述乐曲数据被划分为至少两个乐曲节拍，所述乐曲节拍包括乐曲音符；

信息提取模块，用于从所述乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息，所述节拍数据信息用于描述所述至少两个乐曲节拍的变化速度，所述和弦数据信息用于描述以所述乐曲节拍为单位提取的和弦单元，所述旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况；

数据生成模块，用于根据所述旋律数据信息、所述节拍数据信息以及所述和弦数据信息，

生成乐曲伴奏数据，所述旋律数据信息用于以所述乐曲数据的乐曲旋律限定所述乐曲伴奏的伴奏旋律，所述乐曲伴奏数据用于描述所述乐曲数据的伴奏情况；

伴奏生成模块，用于基于所述乐曲伴奏数据执行音频数据渲染，得到所述乐曲数据对应的所述乐曲伴奏。

另一方面，提供了一种计算机设备，所述计算机设备包括处理器和存储器，所述存储器中存储有至少一段程序，所述至少一段程序由所述处理器加载并执行以实现如上述本申请实施例中任一所述乐曲伴奏的生成方法。

另一方面，提供了一种计算机可读存储介质，所述存储介质中存储有至少一段程序，所述至少一段程序由处理器加载并执行以实现如上述本申请实施例中任一所述的乐曲伴奏的生成方法。

另一方面，提供了一种计算机程序产品或计算机程序，该计算机程序产品包括计算机程序，该计算机程序存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机程序，处理器执行该计算机程序，使得该计算机设备执行上述实施例中任一所述的乐曲伴奏的生成方法。

本申请实施例提供的技术方案带来的有益效果至少包括：

从获取的乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息；以旋律数据信息生成限制条件，通过节拍数据信息和和弦数据信息生成乐曲伴奏数据，最后基于乐曲伴奏数据渲染得到乐曲伴奏。在通过乐曲数据生成乐曲伴奏数据的过程中，通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理，提升对乐曲数据的分析精准性，使得伴奏数据生成过程更加具有针对性。另外，综合旋律数据信息、节拍数据信息以及和弦数据信息，在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现，使得通过乐曲伴奏数据渲染得到的乐曲数据的乐曲伴奏更加精准，提高乐曲伴奏的生成稳定性以及生成效果。

附图说明

- 图1是本申请一个示例性实施例提供的实施环境示意图；
- 图2是本申请一个示例性实施例提供的乐曲伴奏的生成方法的流程图；
- 图3是本申请另一个示例性实施例提供的乐曲伴奏的生成方法的流程图；
- 图4是本申请再一个示例性实施例提供的乐曲伴奏的生成方法的流程图；
- 图5是本申请又一个示例性实施例提供的乐曲伴奏的生成方法的流程图；
- 图6是本申请一个示例性实施例提供的乐曲伴奏的生成方法整体框架示意图；
- 图7是本申请一个示例性实施例提供的伴奏生成方法的输入示意图；
- 图8是本申请一个示例性实施例提供的乐曲伴奏的生成方法的处理流程图；
- 图9是本申请一个示例性实施例提供的钢琴简谱示意图；
- 图10是本申请一个示例性实施例提供的乐曲伴奏的生成装置的结构框图；
- 图11是本申请一个示例性实施例提供的计算机设备的结构框图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作进一步地详细描述。

首先，针对本申请实施例中涉及的名词进行简单介绍。

符号域：在音乐理论中，符号域指的是计算机在处理音乐时使用的符号或数值的范围，也即：符号域指代生成过程中计算机读取、处理的音乐格式为数字音乐格式。在数字音乐格

式中，音符、音高、音量等音乐元素通常以数字形式进行表示。例如：数字音乐格式实现为数字音乐接口（Musical Instrument Digital Interface, MIDI）格式、音乐可标记扩展语言（Music Extensible Markup Language, MusicXML）格式等。例如，MIDI 使用数字表示音符、音高、音量和其他音乐参数。在 MIDI 文件中，不同的数值代表不同的音符，而音符的强度（velocity）可以用数字来表示。符号域在这里可能包括了诸如音符值（表示音符时值的数字，如：4 代表四分音符，8 代表八分音符等）、音高（使用数字或编码表示音符的音高，在 MIDI 中每个音符都有一个对应的音高值）、音量（以数字形式表示音符或乐器的音量级别，较大的数值通常表示更高的音量）等。这些数字或符号构成了计算机在音乐生成过程中所使用的符号域，通过这些符号，计算机能够读取、处理和生成音乐。在这个背景下，符号域的定义涉及到数字表示音乐元素的范围和方式。

钢琴翻弹：指将音频格式的带人声演唱和完整伴奏的原曲用钢琴曲的形式进行改编并演奏。

相关技术中，若希望基于音乐生成对应的音乐伴奏，通常采用基于深度学习的方法进行伴奏生成，借助样本音乐与对应的样本伴奏对模型进行训练，以使得模型学习到生成伴奏的方法；之后将需要生成伴奏的音乐输入训练后的模型，得到音乐伴奏。上述过程过分依赖于样本音乐与样本伴奏之间准确的对应关系，该种方式虽然能够在一定程度上得到良好的音乐伴奏，但是忽略了音乐本身具有的乐理信息，容易导致音乐伴奏的乐曲风格与音乐本身存在较大差距，影响音乐伴奏在音乐领域的准确性。

在本申请实施例中，介绍了一种乐曲伴奏的生成方法，能够通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理，提升对乐曲数据的分析精准性；进而综合至少两个量化信息，在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现，提高乐曲伴奏的生成稳定性以及生成效果。乐曲伴奏的生成方法可以应用于音乐制作领域、音乐创作领域、游戏制作领域、广告制作领域、虚拟现实（Virtual Reality, VR）领域和增强现实（Augmented Reality, AR）领域等，本申请实施例对此不加以限定。

需要说明的是，本申请所涉及的信息（包括但不限于用户设备信息、用户个人信息等）、数据（包括但不限于用于分析的数据、存储的数据、展示的数据等）以及信号，均为经用户授权或者经过各方充分授权的，且相关数据的收集、使用和处理需要遵守相关地区的相关法律法规和标准。例如，本申请中涉及到的乐曲数据、乐曲信息提取等内容都是在充分授权的情况下获取的。

其次，对本申请实施例中涉及的实施环境进行说明，本申请实施例提供的乐曲伴奏的生成方法可以由终端设备单独执行实现，也可以由服务器执行实现，或者由终端设备和服务器通过数据交互实现，本申请实施例对此不加以限定。在一些实施例中，以终端设备和服务器交互执行乐曲伴奏的生成方法为例进行说明。

在一些实施例中，请参考图 1，该实施环境中涉及终端设备 110、服务器 120，终端设备 110 和服务器 120 之间通过通信网络 130 连接。

在一些实施例中，终端设备 110 具有乐曲数据采集功能，能够通过录制、人工制作等方式获取得到乐曲数据。例如：将一首歌作为乐曲数据。

在一些实施例中，终端设备 110 将乐曲数据通过通信网络 130 发送至服务器 120，服务器 120 获取得到乐曲数据。其中，乐曲数据被划分为至少两个乐曲节拍，乐曲节拍包括乐曲音符。

在一些实施例中，服务器 120 从乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息。

其中，节拍数据信息用于表征至少两个乐曲节拍的变化速度，和弦数据信息用于描述以

乐曲节拍为单位提取的和弦单元，旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况。

在一些实施例中，服务器 120 以旋律数据信息为乐曲伴奏的生成限制条件，通过节拍数据信息和和弦数据信息生成乐曲伴奏数据。

其中，旋律数据信息用于以乐曲数据的乐曲旋律限定乐曲伴奏的伴奏旋律，乐曲伴奏数据用于描述乐曲数据的伴奏情况。旋律数据信息作为展现乐曲数据整体的乐曲旋律内容，有利于在乐曲伴奏生成过程中起到整体限定效果，从而在乐曲旋律限定伴奏旋律的基础上，通过节拍数据信息表征的节拍速度以及和弦数据信息表征的局部乐曲音符情况，生成精准性更高的乐曲伴奏数据。

在一些实施例中，服务器 120 基于乐曲伴奏数据执行音频数据渲染，得到乐曲数据对应的乐曲伴奏。

其中，乐曲伴奏数据作为通过音乐参数描述的伴奏情况，是以符号域形式展现的音乐数据内容，因此若需要获取乐曲数据对应的乐曲伴奏，需要对乐曲伴奏数据执行音频数据渲染，从而得到播放的乐曲伴奏。例如：通过特定的音频数据解码软件对乐曲伴奏数据进行分析，从而得到对乐曲伴奏数据进行音频数据呈现的乐曲伴奏。

在一些实施例中，服务器 120 将乐曲伴奏数据通过通信网络发送至终端设备 110，从而由终端设备 110 对乐曲伴奏数据执行音频数据渲染，得到乐曲伴奏；或者，服务器对乐曲伴奏数据执行音频数据渲染，得到便于终端设备进行直接播放的伴奏数据文件，并将伴奏数据文件通过通信网络发送至终端设备 110，终端设备 110 基于伴奏数据文件进行下载并播放乐曲伴奏等。

值得注意的是，上述终端设备包括但不限于手机、平板电脑、便携式膝上笔记本电脑、智能语音交互设备、智能家电、车载终端设备等移动终端设备，也可以实现为台式电脑等；上述服务器可以是独立的物理服务器，也可以是至少两个物理服务器构成的服务器集群或者分布式系统，还可以是云服务器。

在一些实施例中，上述服务器还可以实现为区块链系统中的节点。

本申请中的计算机设备可以是上述终端设备 110、服务器 120 中的至少之一，本申请对此不作限定。下述实施例中各个步骤的执行主体均为计算机设备。

结合上述名词简介和应用场景，对本申请提供的乐曲伴奏的生成方法进行说明，以该方法应用于上述计算机设备为例进行说明，如图 2 所示，该方法包括如下步骤 210 至步骤 240 中的至少一个步骤。

步骤 210，获取乐曲数据。

在一些实施例中，乐曲数据是指通过音频形式表现的数据，用于包含音乐的实际音频内容。

在一些实施例中，乐曲数据可以实现以波形音频文件格式（Waveform Audio File Format, WAV）、动态影像专家压缩标准音频层面（Moving Picture Experts Group Audio Layer III, MP3）、自由无损音频压缩编码（Free Lossless Audio Codec, FLAC）等格式存储，基于其中包含声波信号，因此乐曲数据可用于播放音乐内容。

例如：乐曲数据实现为一首人声歌曲，如：歌手演唱的专辑音乐，或者个人演唱的歌曲，或者个人哼唱的歌曲等；乐曲数据还可以实现为广播电台的节目、电视节目、新闻广播等；乐曲数据还可以实现为音乐会录音或者演唱会录音；乐曲数据还可以实现为电话录音等。

其中，乐曲数据被划分为至少两个乐曲节拍，乐曲节拍包括乐曲音符。

在一些实施例中，乐曲节拍是音乐中的一个基本组织元素，乐曲节拍决定了乐曲音符在时间上的排列和强调情况，作为能够表达音乐时间感的内容，一首音乐通常由强拍和弱拍交替组成。乐曲节拍的强弱关系创建了乐曲数据的节奏感，在不同的音乐风格和文化中，乐曲

节拍的使用方式和强调可能各不相同，乐曲节拍是创作者表达音乐情感和传达音乐信息的重要手段之一。

在一些实施例中，乐曲节拍是拍子中的一个时间单位，每个拍子都有一个强拍（通常是第一拍），其他拍子则被称为弱拍。拍子作为乐谱上表示音乐节奏的符号，由两个数字组成，上面的数字表示每小节中拍子的数量，下面的数字表示每个拍子的时值，时值代表乐曲节拍中乐曲音符的持续时长。例如：4/4 拍子表示每小节有 4 个拍子，每个拍子被称为一个四分音符的时间长度。

也即：乐曲节拍是音乐中一个基本的时间单位，包括在其中的乐曲音符决定了这个时间单位内的音乐信息。在一个拍子内，可以容纳不同时值的乐曲音符，这些音符的组合和排列形成了乐曲的节奏。

步骤 220，从乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息。

在一些实施例中，乐曲信息提取用于对乐曲数据进行量化处理，以将通过音频形式展现的乐曲数据以数据形式进行展现。在一些实施例中，通过乐曲信息提取的方式，从乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息。在一些实施例中，通过对乐曲数据进行量化处理的方式，获取该乐曲数据分别对应的节拍数据信息、和弦数据信息以及旋律数据信息。当然，提取的方式不限于量化处理，还可以通过训练后的神经网络模型来提取节拍数据信息、和弦数据信息以及旋律数据信息。

在一些实施例中，对乐曲数据进行乐曲信息提取，能够获取表征乐曲数据节奏变化的节拍数据信息，还能够获取表征乐曲数据中不同和弦的和弦数据信息，还能够获取表征乐曲数据整体旋律变化的旋律数据信息。

也即：节拍数据信息用于表征至少两个乐曲节拍的变化速度，和弦数据信息用于描述以乐曲节拍为单位提取的和弦单元，旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况。

在一些实施例中，乐曲节拍作为乐曲数据中的基本时间单位，创造了乐曲数据的基本节奏结构，并定义了乐曲音符在时间轴上的排序情况。乐曲节拍在由拍子（Time Signature）表示时，指定每小节中有多少拍子以及每个拍子的时值。在获取节拍数据信息时，可以首先确定乐曲数据对应的至少两个乐曲节拍，进而通过至少两个乐曲节拍的分布变化情况确定节拍数据信息。

在一些实施例中，和弦是同时演奏的至少两个音符的集合，通过以固定的音程关系进行排列。和弦是音乐中的和声基础，对于创造以及强调和谐感十分重要。和弦可以包含基音、第三音、第五音等乐曲音符，从而形成不同的和声质感。也即：和弦关注同时发生的音的组合，其侧重在垂直方向上表示音高关系。在获取和弦数据信息时，可以以乐曲节拍为单位，确定每个乐曲节拍中的乐曲音符，进而将乐曲音频分组为和弦（或称和弦单元）。

在一些实施例中，旋律是一系列有机联系的音符，按照时间的顺序演奏，形成具有旋律线索的音乐表达。旋律是音高和时值的组合，是音乐中最易识别和记忆的部分，通常由主旋律和次旋律组成。也即：旋律侧重关注的是水平方向上的音高和时值的组合。在获取旋律数据信息时，侧重于关注至少两个乐曲音符分别对应的音高变化情况以及至少两个乐曲音符分别对应的时值，从而综合确定与乐曲数据对应的旋律数据信息。

在一个乐曲数据中，节拍、和弦以及旋律这三个元素通常相互交织，共同创造出复杂而有趣的音乐体验。通过乐曲节拍提供了时间框架，通过和弦提供了和声基础，通过旋律给予乐曲引导线索，从而对乐曲数据进行更细致地呈现。

借助乐曲信息提取过程，将乐曲节拍量化为节拍数据信息，将和弦量化为和弦数据信息，将旋律量化为旋律数据信息，以便基于信息提取后的至少两个量化信息对乐曲数据进行分析。

步骤 230，根据旋律数据信息、节拍数据信息和以及和弦数据信息，生成乐曲伴奏数据。

在一些实施例中，在得到旋律数据信息后，基于旋律数据信息表征乐曲数据的整体旋律

情况，因此以旋律数据信息为生成限制条件，以在旋律数据信息表征的乐曲旋律内生成乐曲数据对应的乐曲伴奏。

也即：旋律数据信息用于以乐曲数据的乐曲旋律限定乐曲伴奏的伴奏旋律。

在一些实施例中，乐曲旋律用于表征一系列有机联系的乐曲音符，按照时间的顺序演奏以形成具有旋律线索的音乐表达，属于乐曲数据中最易识别和记忆的部分；旋律数据信息是指表示旋律的数字、符号或计算机可读的形式，可以包括音高序列、时值信息、音符持续时间等，通常以数字或符号的形式进行编码，是用于计算机分析、音乐生成或其他数字音乐处理任务的信息。也即：旋律数据信息与乐曲旋律存在对应关系，旋律数据信息是乐曲旋律的数字形式。

同理，伴奏旋律用于表征一系列有机联系的乐曲音符，按照时间的顺序演奏以形成具有旋律线索的乐曲伴奏。因此，在以旋律数据信息为乐曲伴奏的生成限制条件时，通过旋律数据信息对应的乐曲旋律限定乐曲伴奏对应的伴奏旋律。

其中，乐曲伴奏数据通过音乐参数描述乐曲数据的伴奏情况。这里的音乐参数是指与音乐有关的参数，如音调、音准、音强等等中的至少之一。

在一些实施例中，乐曲伴奏数据与伴奏旋律存在对应关系，乐曲伴奏数据是伴奏旋律的数字形式；在通过节拍数据信息和和弦数据信息进行生成时，生成与伴奏旋律对应的乐曲伴奏数据，该乐曲伴奏数据作为与伴奏旋律对应的数字形式，是通过众多音乐参数对乐曲数据的伴奏进行描述后生成的内容。

基于乐曲伴奏不仅与乐曲数据对应的乐曲旋律相关，也与和弦与节拍相关，因此在生成乐曲伴奏数据时，不仅会通过旋律相关的音高、时值、音色等音乐参数表达，还会通过与和弦相关的音高、和声等音乐参数相关，也会通过与节拍相关的音量、节奏模式等音乐参数相关。也即：用于描述乐曲数据伴奏情况的音乐参数包括音高、音色、音量、节奏模式、和声部分等多种与乐曲伴奏相关的参数内容。

步骤 240，基于乐曲伴奏数据执行音频数据渲染，得到乐曲数据对应的乐曲伴奏。

在一些实施例中，在得到乐曲伴奏数据后，基于乐曲伴奏数据是乐曲伴奏对应的数字形式，因此可以对乐曲伴奏数据进行数据解码，以将乐曲伴奏数据以音频形式进行呈现。也即：对乐曲伴奏数据进行音频数据渲染，从而得到与乐曲数据对应的乐曲伴奏。

在一些实施例中，乐曲伴奏数据实现为描述乐曲数据的数字表示——MIDI 文件形式，在需要获取乐曲伴奏数据对应的乐曲伴奏时，基于 MIDI 文件包含音符、音高、时值等音乐信息，但本身并不包含声音波形，因此需要将 MIDI 文件转换为用于渲染得到乐曲伴奏的伴奏数据。也即：首先基于乐曲伴奏数据进行音频转换，得到伴奏音频；针对伴奏音频进行音频渲染，得到乐曲伴奏。

在一些实施例中，对乐曲伴奏数据进行音频转换并得到伴奏数据的方式包括如下步骤。

(1) 软件工具：借助使用专门的音乐制作软件或 MIDI 编辑器将 MIDI 文件转换为伴奏数据。MIDI 编辑器实现为专业音序器 (Ableton Live)、果味循环工作室 (Fruity Loops Studio, FL Studio)、逻辑增强器 (Logic Pro)、流体合成器 (Fluid Synth)、Timidity++ 等。

(2) 虚拟乐器或音源：在上述软件中选择或加载虚拟乐器或音源插件。这些插件充当虚拟乐器，能够根据 MIDI 文件的信息生成相应的音频波形。

(3) 连接 MIDI 文件：将 MIDI 文件导入到软件中，通常通过拖放文件或导入功能。MIDI 文件包含了音乐的音符、音高、时值等信息。

(4) 分配音色：为 MIDI 文件或音符分配合适的音色 (instrument)，确定使用何种乐器模拟 MIDI 文件中的音符。该过程通常在虚拟乐器或音源插件的界面中完成。

(5) 渲染为音频：借助预设软件的“导出”或“渲染”功能，将 MIDI 文件渲染为音频文件，通常是 WAV、MP3、音频文件格式 (Audio Interchange File Format, AIFF) 等；预设软件会根据 MIDI 文件生成相应的音频波形。

(6) 调整效果：根据需要可以在软件中添加音频效果、混响、均衡器等，以调整音频的最终效果。

值得注意的是，上述将 MIDI 文件转换至伴奏音频的转换过程中，音质和表现力取决于使用的虚拟乐器或音源的质量。一些专业的虚拟乐器库提供高质量的音色和逼真的音频生成；借助上述流程能够将 MIDI 文件中的音乐信息转换为计算机可识别的音频格式（伴奏音频），从而实现在计算机或其他音频设备上打开伴奏音频，以实现播放音乐的目的。

在一些实施例中，对音频转换后的伴奏音频进行渲染，以得到乐曲伴奏。例如：通过终端设备打开伴奏音频，从而播放乐曲伴奏等。

值得注意的是，以上仅为在一些实施例中举例，本申请实施例对此不加以限定。

综上所述，从获取的乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息；以旋律数据信息生成限制条件，通过节拍数据信息和和弦数据信息生成乐曲伴奏数据，最后基于乐曲伴奏数据渲染得到乐曲伴奏。在通过乐曲数据生成乐曲伴奏数据的过程中，通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理，提升对乐曲数据的分析精准性，使得伴奏数据生成过程更加具有针对性。另外，综合旋律数据信息、节拍数据信息以及和弦数据信息，在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现，使得通过乐曲伴奏数据渲染得到的乐曲数据的乐曲伴奏更加精准，提高乐曲伴奏的生成稳定性以及生成效果。

在一些实施例中，借助至少两个乐曲节拍在单位时间内的数量确定节拍数据信息，通过旋律轨和和弦轨获取乐曲数据分别对应的旋律数据信息和和弦数据信息，从而提高数据获取效率和准确性。在一些实施例中，如图 3 所示，上述图 2 所示出的步骤 220 还可以实现为如下步骤 310 至步骤 330 中的至少一个步骤。

步骤 310，基于至少两个乐曲节拍在单位时间内的数量，确定节拍数据信息。

在一些实施例中，单位时间是预先设定的单位时长。例如：单位时间为 1 分钟；或者，单位时间为半分钟；或者，单位时间为 5 秒钟等。

在一些实施例中，以每分钟节拍数（Beat Per Minute, BPM）作为乐曲数据对应的数据节拍信息；BPM 是音乐中用于表示节奏速度的单位，用于表征每分钟内有多少个拍子。在确定 BPM 的过程中，可以采用手动计算的方式，也可以依赖音乐制作软件等工具。例如：采用手动计算方式时，如果能感受到音乐的节奏则可以使用手表或计时器在一分钟内确定拍子的次数，如在一分钟内数到 120 下则 BPM 为 120；或者，采用音乐节拍检测器确定 BPM，可以显示歌曲的 BPM 等。

步骤 320，提取乐曲数据对应的和弦轨和旋律轨。

其中，和弦轨用于描述具有时序关系的至少两个乐曲音符以和弦形式构建的音群。

在一些实施例中，和弦轨（Chord Track）是音乐制作软件中的一个功能，用于表示音乐中的和弦进度。

在一些实施例中，和弦轨通常会被分割成小节，以便更好地组织和理解音乐的结构。和弦轨的小节的数量取决于音乐的结构和编排；在音乐制作软件中，可以在和弦轨上创建和编辑和弦，每个小节通常对应音乐中的一个小节，从而有助于在整个曲目中保持一致的和弦进度，便于调整和组织音乐。

其中，旋律轨用于描述至少两个乐曲音符依照时序关系构成的主旋律。

在一些实施例中，在音乐制作软件中，旋律轨（Melody Track）是音乐制作中的一个概念，它指的是包含主旋律的音轨或轨道。主旋律是音乐中最突出、最引人注目的旋律线，通常由乐器或歌唱（如人声歌唱）演出。在音乐制作软件中，制作者可以使用旋律轨录制、编辑和排列主旋律。

其中，旋律轨作为用于容纳主旋律的音轨，可以将乐曲数据中最重要的、最易记的旋律

线进行记录。因为主旋律是音乐中吸引听众注意力的关键元素之一。通过旋律轨使得制作者可以精确地塑造和控制音乐的核心元素，从而创造出具有独特风格和情感的作品。

在一些实施例中，和弦轨通常也会被分割成小节。旋律轨小节的数量取决于音乐的结构、编曲和制作者的创作风格。

步骤 330，基于旋律轨和和弦轨，确定旋律数据信息和和弦数据信息。

在一些实施例中，以旋律轨中小节的数量为基准，调整和弦轨中小节的数量，得到和弦数据信息；基于旋律轨中至少两个乐曲音符分别对应的音符音高，对旋律轨执行旋律变换，得到旋律数据信息。

在一些实施例中，分别确定旋律轨中小节的数量以及和弦轨中小节的数量，得到旋律轨对应的第一数量以及和弦轨对应的第二数量。

其中，第一数量用于表征旋律轨中小节的数量，即：旋律轨中包括第一数量的小节；第二数量用于表征和弦轨中小节的数量，即：和弦轨中包括第二数量的小节。

在一些实施例中，旋律轨和和弦轨通常均会被各自分割成小节；将旋律轨分割为小节，得到第一数量的小节；将和弦轨分割为小节，得到第二数量的小节。

在一些实施例中，预先获取音乐特征提取模型，音乐特征提取模型是预先训练的模型，用于对乐曲数据进行诸如乐曲节拍、和弦、旋律等特征的分析过程。借助音乐特征提取模型对旋律轨和和弦轨分别进行小节提取，从而得到旋律轨中第一数量的小节，以及得到和弦轨中第二数量的小节。

在一些实施例中，第一数量和第二数量可能相等；或者，第一数量和第二数量不同，如：第一数量小于第二数量，或者第二数量小于第一数量等。

在一些实施例中，以第一数量为基准，对和弦轨的第二数量进行调整，得到调整后的和弦轨作为和弦数据信息。

在一些实施例中，响应于第二数量小于第一数量，基于第一数量对和弦轨进行补齐操作，得到补齐后的和弦轨作为和弦数据信息。补齐后的和弦轨即为上述调整后的和弦轨。

在一些实施例中，若第二数量小于第一数量，则代表对和弦轨进行小节划分后得到小节的数量较少，对旋律轨进行小节划分后得到小节的数量较多；基于旋律轨能够更好地表征乐曲数据的整体效果，因此以旋律轨对应的第一数量为准，对第二数量进行补齐操作，从而得到补齐后的和弦轨作为和弦数据信息。

在一些实施例中，在补齐操作过程中，通过复制和弦轨中需要补齐小节的前一小节的和弦，得到补齐后的和弦轨作为和弦数据信息。

在一些实施例中，对补齐后的和弦轨中的每一小节进行检测，为避免伴奏有过长时间的休止，当存在小节没有任何和弦时，复制上一小节的和弦；若该小节为第一小节，则复制后面第一个不为全休止的小节的和弦。

在一些实施例中，响应于第二数量等于第一数量，将和弦轨作为和弦数据信息；响应于第二数量大于第一数量，将旋律轨与和弦轨进行对齐操作，得到与旋律轨对应的对齐和弦轨作为和弦数据信息等。

在一些实施例中，基于旋律轨中至少两个乐曲音符分别对应的音符音高，对旋律轨进行旋律变换处理，得到更新后的旋律轨作为旋律数据信息。

在一些实施例中，旋律轨描述了至少两个乐曲音符分别对应的音符特征，音高是音乐中一个基本而重要的概念，它决定了乐曲音符的相对高低。

在一些实施例中，对于部分歌曲旋律普遍音高较低的情况，为提升听感同时防止与伴奏冲突，可以选择性地对旋律进行高八度处理。

本申请实施例提供的技术方案，通过旋律轨中小节的数量来调整和弦轨中的小节的数量，实现了对于和弦轨的补齐，最终得到补齐后的和弦轨对应的和弦数据信息。通过旋律轨中乐曲音符的音符音高，对旋律轨执行旋律变换处理，使得旋律更加和谐，最终得到了旋律数据

信息。此种方式有利于保证获取到的和弦数据信息和旋律数据信息的准确性。

在一些实施例中，获取预设的第一音高和第二音高，第一音高和第二音高用于限定对旋律轨执行旋律变换的变换方式。示例性地，第一音高为预设低音高，第二音高为预设高音高。也即，获取预设低音高和预设高音高。示例性地，第一音高和第二音高对应不同的音高数值，第一音高低于第二音高。

其中，预设低音高和预设高音高用于综合限定对旋律轨执行旋律变换处理的处理方式。

在一些实施例中，将旋律轨中至少两个乐曲音符分别对应的音符音高，分别与第一音高和第二音高比较，确定第一音符比例和第二音符比例，第一音符比例用于表示第一音符占至少两个乐曲音符的音符比例，第一音符用于表示低于第一音高的乐曲音符，第二音符比例用于表示第二音符占至少两个乐曲音符的音符比例，第二音符用于表示高于第二音高的乐曲音符；基于第一音符比例和第一音符比例，对旋律轨执行旋律变换，得到旋律数据信息。

在一些实施例中，第一音符比例在下述实施例中也称为低音音符比例。在一些实施例中，第二音符比例在下述实施例中也称为高音音符比例。

在一些实施例中，第一音符在下述实施例中也称为低音音符。在一些实施例中，第二音符在下述实施例中也称为高音音符。

在一些实施例中，将旋律轨中至少两个乐曲音符分别对应的音符音高与预设低音高进行比较，以及将至少两个乐曲音符分别对应的音符音高与预设高音高进行比较，确定低音音符比例以及高音音符比例。

其中，低音音符比例用于表示低音音符占至少两个乐曲音符的音符比例，低音音符用于表示低于预设低音高的乐曲音符；高音音符比例用于表示高音音符占至少两个乐曲音符的音符比例，高音音符用于表示高于预设高音高的乐曲音符。

在一些实施例中，基于低音音符比例以及高音音符比例对旋律轨执行旋律变换处理，得到更新后的旋律轨作为旋律数据信息。

在一些实施例中，在获取得到旋律轨后，确定其中至少两个乐曲音符分别对应的音符音高；获取预设低音高和预设高音高，将至少两个乐曲音符分别与预设低音高和预设高音高进行比较，以分别计算低音音符比例以及高音音符比例。

例如：以预设低音高为 56，预设高音高为 70 为例，在确定至少两个乐曲音符分别对应的音符音高后，将至少两个音符音高与 56 和 70 分别进行比较，若乐曲音符对应的音符音高低于 56，视为该乐曲音符为低音音符；若乐曲音符对应的音符音高高于 70，视为该乐曲音符为高音音符；从而能够确定低音音符的第一音符数量和高音音符的第二音符数量，进而综合第一音符数量与乐曲音符的乐曲音符数量确定低音音符比例，综合第二音符数量与乐曲音符数量确定高音音符比例。

在一些实施例中，在低音音符比例和高音音符比例同时符合预设条件的情况下，不对旋律轨进行调整。

例如：当高音音符比例超过 10%且低音音符比例低于 20%时，则不对旋律轨进行高八度处理；当不符合上述预设条件的情况，对旋律轨进行调整。

在一些实施例中，为了使得乐曲有完整度和终止感，可以为每首乐曲加上尾奏，长度范围为一个小节到三个小节。由于不同乐曲旋律结尾小节不同，有的结尾为一个长音，有的结尾有长时间的休止，有的则没有足够的时间来加尾奏。因此需要借助旋律轨的情况，提前判断是否对伴奏轨加上尾奏小节。

在一些实施例中，当旋律轨的结尾小节的最后一个音符结束在第二拍以后，且不是长音情况，则在生成伴奏轨时，预先加上一个空白的尾奏小节，且和弦复制上一小节的和弦；从而在对尾奏进行织体转换时，如果选择的织体超过四拍，则会在写入 MIDI 时扩展至两到三个小节。

在一些实施例中，借助上述过程基于旋律轨和和弦轨获取得到旋律数据信息以及和弦数

据信息，该旋律数据信息和和弦数据信息用于综合节拍数据信息对乐曲数据进行数字形式的表达。

本申请实施例提供的技术方案，通过预设低音高和预设高音高，与旋律轨中至少两个乐曲音符分别对应的音符音高的进行比较，从而得出乐曲音符中高音音符比例和低音音符比例。根据音符比例的不同，采用不同的旋律处理方式，体现了旋律处理的灵活性和有针对性，从而提高了旋律数据信息的确定效率和确定精度。

值得注意的是，以上仅为在一些实施例中举例，本申请实施例对此不加以限定。

综上所述，从获取的乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息；以旋律数据信息生成限制条件，通过节拍数据信息和和弦数据信息生成乐曲伴奏数据，最后基于乐曲伴奏数据渲染得到乐曲伴奏。在通过乐曲数据生成乐曲伴奏数据的过程中，通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理，提升对乐曲数据的分析精准性，使得伴奏数据生成过程更加具有针对性。另外，综合旋律数据信息、节拍数据信息以及和弦数据信息，在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现，使得通过乐曲伴奏数据渲染得到的乐曲数据的乐曲伴奏更加精准，提高乐曲伴奏的生成稳定性以及生成效果。

在一些实施例中，首先获取乐曲数据中至少两个乐曲小节分别对应的伴奏织体，进而根据伴奏织体对对应乐曲小节进行调整，从而综合至少两个调整后的乐曲小节组合生成乐曲伴奏数据。在一些实施例中，如图4所示，上述图2所示出的步骤230还可以实现为如下步骤410至步骤430。

步骤410，在以乐曲小节为单位的情况下，以旋律数据信息为乐曲伴奏的生成限制条件，基于节拍数据信息和和弦数据信息，确定与至少两个乐曲小节分别对应的伴奏织体。

其中，乐曲小节又称小节，其中包括预设数量的乐曲节拍，是对乐曲数据进行细化分析时的一种基本组织单元，用于划分和组织音乐片段；引入小节对乐曲数据进行分析，便于提升对乐曲小节的理解程度。

以乐曲小节为单位用于表示对乐曲数据的每个乐曲小节分别进行分析。针对每个乐曲小节，以旋律数据信息为乐曲伴奏的生成限制条件，从而综合基于节拍数据信息和和弦数据信息确定伴奏织体。

其中，伴奏织体是用于衬托旋律数据信息对应的主旋律的声音元素。在一些实施例中，伴奏织体是指乐曲中负责支持和衬托主旋律的声音层，这一层通常包括和弦、低音、鼓点、打击乐等元素，它们的组合形成了音乐的基础和背景。伴奏织体在音乐中的作用是为主旋律提供支持，增添音乐的层次和丰富度。合理运用伴奏织体可以使音乐更加完整、丰富，并增强听众的音乐体验。不同类型的音乐可能会有不同的伴奏织体构建方式，以适应不同的风格和情感表达。

在一些实施例中，确定与至少两个乐曲小节分别对应的伴奏织体，以通过伴奏织体对该乐曲小节内的乐曲音符进行调整，从而丰富该乐曲小节的表现形式。进一步地，组合乐曲小节对应的调节小节所得到的乐曲伴奏数据也更加多样，体现了乐曲伴奏的生成方式的多样性和灵活性。

在一些实施例中，在确定伴奏织体时，综合旋律、节拍以及和弦进行统筹调整，以确定便于提升该乐曲小节音乐表现效果的伴奏织体。

在一些实施例中，在以乐曲小节为单位的情况下，获取织体确定规则。

其中，织体确定规则用于通过乐曲信息提取结果确定与乐曲小节对应的伴奏织体。

在一些实施例中，乐曲信息提取结果用于表示通过乐曲信息提取过程提取得到的结果，乐曲信息提取结果包括旋律数据信息、节拍数据信息以及和弦数据信息中的至少之一。

也即：获取的织体确定规则用于综合乐曲小节内的旋律数据信息、节拍数据信息以及和

弦数据信息，确定与该乐曲小节对应的伴奏织体。

在一些实施例中，在织体确定规则的限定下，通过旋律数据信息限定乐曲伴奏的音符变化幅度，通过节拍数据信息限定乐曲伴奏的节奏变化情况，通过和弦数据信息限定乐曲伴奏的音符协调程度，确定与乐曲小节分别对应的伴奏织体。

在一些实施例中，旋律数据信息与乐曲音符息息相关，因此可以通过旋律数据信息辅助确定乐曲伴奏中伴奏音符的音高、变化等，也即：可以通过旋律数据信息限定乐曲伴奏的音符变化幅度。

在一些实施例中，旋律数据信息用于通过音高范围、音高密度以及音高值（如最低音的音高值），限定乐曲伴奏对应的音符变化幅度。

在一些实施例中，节拍数据信息与乐曲音符组成的乐曲节拍息息相关，因此可以通过节拍数据信息辅助确定乐曲伴奏中伴奏音符的音符节奏、音符变化速率等情况。

在一些实施例中，节拍数据信息用于通过单位时间内乐曲节拍的数目，确定乐曲伴奏对应的节奏类型。示例性地，节奏类型包括慢速、中速和快速等。

在一些实施例中，和弦数据信息作为通过至少两个音符组成的和弦内容确定的信息，与乐曲音符本身息息相关，因此可以通过和弦数据信息辅助确定乐曲伴奏中伴奏音符的音符形式，即：确定乐曲伴奏中应当采用的音符，从而更细致地提高乐曲伴奏的准确性。

在一些实施例中，和弦数据信息用于通过和弦位置辅助旋律数据信息确定音符协调程度。

本申请实施例中，旋律数据信息用于限定乐曲伴奏对应的音符变化幅度、节拍数据信息用于确定乐曲伴奏对应的节奏类型、和弦数据信息用于确定音符协调程度，基于这三个数据所确定出来的乐曲伴奏与乐曲数据较为贴合，有利于保证伴奏生成效果和生成效率。

在一些实施例中，对每个乐曲小节进行分析，获取得到与每个乐曲小节分别对应的伴奏织体。

在一些实施例中，伴奏织体的分类常见于乐曲中不同声部（例如：和声、低音、打击乐等）的组合，以及它们在乐曲中的作用。伴奏织体包括：和声伴奏（以和弦为基础，负责支持主旋律）、节奏伴奏（以鼓点和打击乐为基础，负责提供节奏感）、低音伴奏（以低音乐器为基础，为音乐提供稳定的低音基础）、旋律伴奏（与主旋律相伴奏，通过和声、音色或音乐装饰来衬托主旋律）等。

也即：根据旋律、和弦以及节拍，综合确定每一个乐曲小节分别对应的伴奏织体类型，以便通过伴奏织体对对应乐曲小节进行调整。

在一些实施例中，以小节为单位进行织体转换时，对每一首歌从前往后循环处理。为了适配不同速度的歌曲，根据 BPM 设计了三套织体转换程序。BPM 不超过 90 判断为慢速，BPM 超过 90 但低于 120 判断为中速，BPM 达到 120 则判断为快速。慢速歌曲的织体有最丰富的节奏类型，而中速歌曲大部分情况下音符最小时值为八分音符，快速歌曲则为四分音符。

速度判断完成后，进入该小节的旋律判断。计算旋律的音高范围、音符密度以及最低音的音高，选择高或低、稀疏或稠密的伴奏织体。在每种情况下，从满足条件的备选中选取伴奏织体也具有一定的随机性。对于第一小节需要判断是否满足为弱起小节且是否加前奏。判断规则为：非弱起小节或第一个音符开始在第二拍之前按正常小节处理，弱起小节但第一个音符出现在第三拍之前不配伴奏，弱起小节但第一个音符出现在第三拍及以后则随机选择不配伴奏或配前奏，前奏只根据第一个和弦来选择伴奏织体。

伴奏织体选择完成后，为避免本小节音符总时值发生错误，从而导致偏差累计，因此可以设置每小节节拍数的限制：在有限的循环次数下，判断当前小节所有音符的时值之和是否等于 4 拍，超过则移除最后一个音符或休止，不足则补上休止。为避免破坏设计好的伴奏织体，需要配前奏的小节和结尾小节没有此限制。

对最后一个伴奏小节，可以首先判断旋律：空小节则直接配尾奏；休止达到三拍且最后一个音结束时间在第一拍及以前，或拖长音，加一拍和弦再配尾奏；其余加两拍和弦再配尾

奏。

值得注意的是，以上仅为在一些实施例中举例，本申请实施例对此不加以限定。

步骤 420，通过伴奏织体，调整伴奏织体对应的乐曲小节内乐曲音符的音符时值，得到与至少两个乐曲小节分别对应的调整小节。

在一些实施例中，获取得到与每个乐曲小节分别对应的伴奏织体后，通过伴奏织体能够对乐曲小节内的乐曲音符进行调整。

在一些实施例中，通过伴奏织体对乐曲音符对应的音符时值进行调整，音符时值用于表示乐曲音符的持续时长。

在一些实施例中，在乐曲小节内修改乐曲音符的时值，使乐曲音符变得更短或更长。该过程可以通过将乐曲音符拆分成较短的音符或将至少两个乐曲音符合并成一个较长的音符实现。例如：将一个四分音符分解为两个八分音符，或将两个八分音符合并为一个四分音符等。

此外，在修改乐曲音符的时值时，还可以使用连音或断音调整音符时值。也即：通过连音将相邻的乐曲音符连接起来，使它们在时间上更紧密；通过断音则在乐曲音符之间创建短暂的间隔，增加一些空白感。

在一些实施例中，还可以通过伴奏织体对乐曲小节中的乐曲音符增加装饰音，如：颤音、滑音、连音等，可以使原有的乐曲音符更具表现力，这些装饰音可以在不改变基本旋律的情况下增加音乐的变化。

在一些实施例中，还可以通过伴奏织体在乐曲小节中引入新的节奏图案，例如：引入更复杂的鼓点、弱拍的强调或交叉节奏，该过程可以通过在伴奏织体中的鼓点或其他打击乐器上进行变化实现。

在一些实施例中，还可以通过伴奏织体在乐曲小节中强调弱拍或拍子的其他部分，使乐曲伴奏更富动感。

在一些实施例中，还可以通过伴奏织体在乐曲小节的末尾加速或减速，从而创造出一种渐进的变化等。

在一些实施例中，上述内容基于乐曲小节对应的伴奏织体确定，伴奏织体借助织体确定规则、旋律、和弦以及节拍综合确定。

其中，调整小节中包括至少两个调整后的乐曲音符。

在一些实施例中，通过伴奏织体对对应乐曲小节中的乐曲音符进行调整，从而得到至少两个调整后的乐曲音符，从而确定每个乐曲小节内调整后的乐曲音符，并将每个调整后的乐曲小节称为调整小节。

步骤 430，组合与至少两个乐曲小节分别对应的调整小节，生成乐曲伴奏数据。

在一些实施例中，确定至少两个乐曲小节在乐曲数据内的时序关系；依照时序关系对至少两个调整小节进行组合，从而生成乐曲伴奏数据。

在一些实施例中，基于调整小节是通过伴奏织体调整乐曲音符得到的，因此调整小节侧重于表达乐曲伴奏的数字信息，由至少两个调整小节生成的乐曲伴奏数据也侧重于表达乐曲伴奏的数字信息。例如：乐曲伴奏数据是以 MIDI 文件形式表达的信息。进一步地，通过织体确定规则、旋律数据信息、节拍数据信息以及和弦数据信息来确定伴奏织体，有利于保证伴奏织体的确定准确性和确定效率，从而提高乐曲伴奏的生成效果。

值得注意的是，以上仅为在一些实施例中距离，本申请实施例对此不加以限定。

综上所述，从获取的乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息；以旋律数据信息生成限制条件，通过节拍数据信息和和弦数据信息生成乐曲伴奏数据，最后基于乐曲伴奏数据渲染得到乐曲伴奏。在通过乐曲数据生成乐曲伴奏数据的过程中，通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理，提升对乐曲数据的分析精准性，使得伴奏数据生成过程更加具有针对性。另外，综合旋律数据信息、节拍数据信息以及和弦数据信息，在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能

够通过音乐参数进行更细致地呈现,使得通过乐曲伴奏数据渲染得到的乐曲数据的乐曲伴奏更加精准,提高乐曲伴奏的生成稳定性以及生成效果。

在一些实施例中,在通过乐曲伴奏数据进行音频数据渲染时,为了能够提高乐曲伴奏数据的旋律准确性,可以将旋律数据信息和乐曲伴奏数据进行混音处理,进而通过对混音结果进行音频数据渲染的过程得到乐曲结果。在一些实施例中,如图5所示,上述图2所示出的实施例还可以实现为如下步骤510至步骤550;上述图2所示出的步骤240还可以实现为如下步骤540至步骤550。

步骤510,获取乐曲数据。

其中,乐曲数据被划分为至少两个乐曲节拍,乐曲节拍中包括乐曲音符。

在一些实施例中,步骤510已经在上述步骤210进行了说明,此处不再赘述。

步骤520,从乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息。

其中,节拍数据信息用于表征至少两个乐曲节拍的变化速度,和弦数据信息用于描述以乐曲节拍为单位提取的和弦单元,旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况。

在一些实施例中,步骤520已经在上述步骤220以及步骤310至步骤330中进行了说明,此处不再赘述。

步骤530,根据旋律数据信息为乐曲伴奏、节拍数据信息以及和弦数据信息,生成乐曲伴奏数据。

其中,旋律数据信息用于以乐曲数据的乐曲旋律限定乐曲伴奏的伴奏旋律,乐曲伴奏数据通过音乐参数描述乐曲数据的伴奏情况。

在一些实施例中,步骤530已经在上述步骤230以及步骤410至步骤430中进行了说明,此处不再赘述。

步骤540,对旋律数据信息和乐曲伴奏数据执行混音处理,得到混音结果。

在一些实施例中,为了在体现伴奏的同时能够更贴切地展现乐曲旋律,将乐曲旋律对应的旋律数据信息与生成的乐曲伴奏数据进行混音处理。

其中,混音是指将不同音频轨道(如:旋律和伴奏)合并在一起,以创造出整体音乐的过程。混音的目的是在保持音频清晰度和平衡的同时,将各个音轨结合成一个完整的音乐作品。将旋律数据信息和乐曲伴奏数据进行混音处理,目的在于创造音乐的整体感,使得乐曲数据本身对应的乐曲旋律数据以及生成的乐曲伴奏数据成为一个有机的整体,使得混音后能够相互衬托而不互相压制,形成均衡、压缩、混响的音频效果等。

在一些实施例中,通过音源混响器对旋律数据信息和乐曲伴奏数据进行混音处理,生成混音结果。

在一些实施例中,旋律数据信息和乐曲伴奏数据均实现为MIDI文件,将混音处理所涉及的旋律数据信息和乐曲伴奏数据合并在一起,创造出一个整体的MIDI文件。这可以在音符、音量、音色等方面进行调整,以创造出更为复杂和丰富的音乐作品。

在一些实施例中,确定旋律数据信息对应的乐曲旋律轨,以及乐曲伴奏数据对应的乐曲伴奏轨;在音频轨对齐的条件下,对乐曲旋律轨和乐曲伴奏轨进行混音处理,生成混音结果。

在一些实施例中,旋律数据信息表征乐曲旋律轨,其中包括乐曲音符基于旋律数据信息的表现情况;乐曲伴奏数据表征乐曲伴奏轨,其中包括乐曲音符基于乐曲伴奏数据的表现情况。音频轨对齐即用于表征乐曲旋律轨和乐曲伴奏轨对齐的条件下,对乐曲旋律轨和乐曲伴奏轨进行混音处理,从而生成混音结果。此种方式,可以保证混音处理的正确性,从而提高生成的混音结果的准确性。

在一些实施例中,混音处理可以实现为如下步骤。

(1) 导入旋律数据信息和乐曲伴奏数据:将两个需要混音的信息导入到MIDI编辑软件

或数字音频工作站（Digital Audio Workstation, DAW）中，确保它们在时间轴上正确对齐。

(2) 调整起始点：确保旋律数据信息和乐曲伴奏数据在时间轴上从相同的起始点开始，这样它们能够同步。

(3) 音符编辑：对旋律数据信息和乐曲伴奏数据中的乐曲音符进行编辑。如删除或调整一些音符，使它们更好地相互融合。也可以考虑将一些音符的时值进行调整，以创造更丰富的和弦或旋律变化。

(4) 音量平衡：调整旋律数据信息和乐曲伴奏数据的音量，确保它们在混合后能够保持平衡，这可以通过调整旋律数据信息和乐曲伴奏数据分别对应的通道上的音量控制器实现。

(5) 音色和表情：在混音过程中，可以调整旋律数据信息和乐曲伴奏数据分别对应的通道上的音色（乐器选择）以及表达力控制器（如音量、音符长度、音色等），以使整体音乐更富有表现力。

(6) 节奏和时间感：调整旋律数据信息和乐曲伴奏数据的节奏和时间感觉，使它们在混合后更好地协调，如：乐曲音符的时间偏移、加入一些人为的微时值变化等。

(7) 效果处理：应用一些变换效果，如：合并效果、延迟效果、合唱效果等，以增加混音的复杂性和层次感。

(8) 混音和导出：在混音过程的最后阶段，确保旋律数据信息和乐曲伴奏数据的混合效果达到预设要求后，导出混音后的混音结果，混音结果实现为 MIDI 文件形式。

步骤 550，对混音结果执行音频数据渲染，得到乐曲数据对应的乐曲结果。

其中，乐曲结果中以乐曲伴奏为伴奏内容。

在一些实施例中，得到混音结果后，对混音结果进行音频数据渲染，以得到表达完整乐曲内容的乐曲结果，基于乐曲结果生成时依赖乐曲伴奏数据，因此该乐曲结果中包括乐曲伴奏数据对应的乐曲伴奏，从而有利于在生成乐曲伴奏的同时通过旋律数据信息更真实地展现乐曲内容。

在一些实施例中，混音数据实现为描述乐曲数据的数字表示——MIDI 文件形式，在需要获取混音数据对应的乐曲结果时，基于 MIDI 文件包含音符、音高、时值等音乐信息，但本身并不包含声音波形，因此需要将 MIDI 文件转换为用于渲染得到乐曲结果的数据。也即：首先基于混音结果进行音频转换，得到乐曲结果音频；针对乐曲结果音频进行音频渲染，得到乐曲结果。

在一些实施例中，对音频转换后的乐曲结果音频进行渲染，以得到乐曲结果。例如：通过终端设备打开乐曲结果音频，从而播放具有乐曲伴奏的乐曲结果等。

值得注意的是，以上仅为在一些实施例中举例，本申请实施例对此不加以限定。

综上所述，从获取的乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息；以旋律数据信息生成限制条件，通过节拍数据信息和和弦数据信息生成乐曲伴奏数据，最后基于乐曲伴奏数据渲染得到乐曲伴奏。在通过乐曲数据生成乐曲伴奏数据的过程中，通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理，提升对乐曲数据的分析精准性，使得伴奏数据生成过程更加具有针对性。另外，综合旋律数据信息、节拍数据信息以及和弦数据信息，在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现，使得通过乐曲伴奏数据渲染得到的乐曲数据的乐曲伴奏更加精准，提高乐曲伴奏的生成稳定性以及生成效果。

在一些实施例中，上述乐曲数据可以实现为任何一首流行歌曲，将乐曲伴奏的生成方法应用于流行音乐的伴奏生成过程中，该方法还可以称为“一种流行歌曲的符号域钢琴翻弹自动生成方法”，如图 6 所示，对本申请实施例的整体框架图。

在一些实施例中，在获取得到原曲音频文件 610（即上述乐曲数据）后，首先采用预设算法（如神经网络 transform）对原曲音频文件 610 进行旋律提取、节拍跟踪以及和弦提取，

以获取音乐信息，其中包括旋律提取后的旋律音符 621、节拍跟踪后的节拍信息 622、和弦提取后的和弦序列 623。

提取出的旋律音符 621 由起点、偏移点和音高表示；将旋律音符 621 再与从原曲音频文件 610 中检测出的节拍信息 622 对齐并量化到十六分音符，得到旋律 MIDI 文件（即上述旋律数据信息）。

和弦提取中以一拍为最小单位，包括大三、小三和弦，得到和弦序列 623。

将旋律与和弦序列删除没有旋律的小节后合成 MIDI 文件，拍号为 4/4 拍。

此外，进一步从节拍信息 622 中估计出原曲音频文件 610 的速度信息，即 BPM，对原曲音频文件 610 的平均速度做轻微降速处理；其次，在以旋律为条件控制伴奏的音域、疏密程度等的同时，伴奏生成算法根据和弦级数、种类等生成与旋律匹配的织体，输出钢琴伴奏 MIDI 文件（即：乐曲伴奏数据）。最后，将旋律、伴奏两轨 MIDI 通过批量化的音乐制作方法，渲染出钢琴翻弹的音频文件（即用于渲染得到乐曲伴奏的乐曲音频）。

在一些实施例中，伴奏生成算法的输入示例如图 7 所示，其中包括至少两个音符 710，输入文件为“.mid”格式，即：MIDI 文件，其中存储旋律轨与和弦轨两轨，包含拍号（均为 4/4 拍）、调号元信息。和弦预处理为具体音高而非“C”、“Am”等和弦标记形式，输入和弦的音高范围在默认音区（从小字一组开始），仅考虑原位大、小三和弦，但织体中包含转位以及七音、九音等。

借助伴奏生成算法执行伴奏生成方法，如图 8 所示，其中主要包括如下五个流程：（1）输入读取 810：输入 MIDI 及 BPM 读取；（2）预处理 820：和弦轨预处理；（3）织体转换 830；（4）后处理 840：旋律及伴奏后处理；（5）写入（MIDI）文件 850。

（1）输入读取 810

在一些实施例中，读取输入 MIDI 文件，将旋律、和弦分两轨存入两个声部，输入估算的原曲 BPM 作为选择伴奏织体的条件之一。

（2）预处理 820

分别计算旋律轨、和弦轨的小节的数量，若和弦轨中小节的数量少于旋律轨中小节的数量，则补齐，补上的小节复制前一小节的和弦。此外对每一小节进行检测，为避免伴奏有过长时间的休止，当有小节没有任何和弦时，算法将复制上一小节的和弦；若该小节为第一小节，则复制后面第一个不为全休止的小节的和弦。

对于部分歌曲旋律普遍音高较低的情况，为提升听感同时防止与伴奏冲突，可以将旋律高八度处理。具体判断方法为以 MIDI 音高 56 和 70 为阈值，分别计算低音音符比例和高音音符比例，当高音音符比例超过 10%且低音音符比例低于 20%时，则不做高八度处理。

为使乐曲有完整度和终止感，可以为每首乐曲都会加上尾奏，长度范围为一个到三个小节。由于不同乐曲旋律结尾小节不同，有的结尾为一个长音，有的结尾有长时间的休止，有的则没有足够的时间来加尾奏。因此需要提前判断是否对伴奏轨加上尾奏小节。具体判断规则为：当旋律结尾小节的最后一个音符结束在第二拍以后，且不是长音情况，则预先加上一个空白的尾奏小节，和弦复制上一小节的和弦。在对尾奏进行织体转换时，如果选择的织体超过四拍，则会在写入 MIDI 时扩展至两到三个小节。

（3）织体转换 830

以小节为单位进行织体转换，对每一首歌从前往后循环处理。为了适配不同速度的歌曲，根据 BPM 设计了三套织体转换程序。BPM 不超过 90 判断为慢速，BPM 超过 90 但低于 120 判断为中速，BPM 达到 120 则判断为快速。慢速歌曲的织体有最丰富的节奏类型，而中速歌曲大部分情况下音符最小时值为八分音符，快速歌曲则为四分音符。

速度判断完成后，进入该小节的旋律判断。计算旋律的音高范围、音符密度以及最低音的音高，选择高或低、稀疏或稠密的伴奏织体。在每种情况下，从满足条件的备选中选取织体也具有一定的随机性。对于第一小节需要判断是否满足为弱起小节且是否加前奏。具体规

则为：非弱起小节或第一个音符开始在第二拍之前按正常小节处理，弱起小节但第一个音符出现在第三拍之前不配伴奏，弱起小节但第一个音符出现在第三拍及以后则随机选择不配伴奏或配前奏，前奏只根据第一个和弦来选择织体。

织体选择完成后，为避免本小节音符总时值发生错误，从而导致偏差累计，因此设置了每小节节拍数的限制：在有限的循环次数下，判断当前小节所有音符的时值之和是否等于 4 拍，超过则移除最后一个音符或休止，不足则补上休止。为避免破坏设计好的织体，需要配前奏的小节和结尾小节没有此限制。

对最后一个伴奏小节，算法首先判断旋律：空小节则直接配尾奏；休止达到三拍且最后一个音结束时间在第一拍及以前，或拖长音，加一拍和弦再配尾奏；其余加两拍和弦再配尾奏。

在一些实施例中，织体类型包括柱式、半分解式、全分解式，节奏型有四分、二八、十六、切分、附点、五连音等多种，并包含琶音等装饰音，具有丰富的多样性。在输入只有大三、小三和弦的情况下，根据旋律的音域及疏密程度，选择性地增加七音、九音、十三音以及和弦转位，扩展和弦类型。在考虑歌曲速度的情况下，各种专门设计的织体可以涵盖抒情慢歌、中速歌曲、快速热歌各种风格的歌曲，同时保持舒缓的整体风格。此外，还针对休止达到三拍的弱起小节设计了前奏的织体，与弱起小节不配伴奏两种情况随机选取。为使乐曲有终止感，还设计了多种尾奏，根据旋律结尾小节的音符时值、结束位置等信息来选取。

织体转换分为函数和文件两种转换方式。函数转换简单稳定可以适用于所有输入和弦类型的情况，而文件转换则更利于扩展。函数转换指有相关的织体转换函数针对小节内各种和弦时值的情况，直接从和弦中选取基本音符，并根据和声学设计音高、时值，组合单音以及和弦，生成完整的织体小节并返回。文件转换指使用数字乐谱文件，即“`.mxl`”文件存储各个小节的织体，按大三和弦、小三和弦两种情况，以及和弦时值为四拍、三拍、两拍、一拍四种情况，每一类织体共分为八个文件。乐谱文件中每个小节存储一种织体，均为 C 或 Cm 和弦及其变体，使用时根据当前和弦根音与 C4 音高的音程关系来移调。如图 9 其中示出至少两个乐曲音符 910，至少两个乐曲音符 910 表征 Cadd9 和弦，时值为 4 拍的织体文件示例。

(4) 后处理 840

完成织体转换后，算法首先将估算的原曲 BPM 插入当前乐谱，除首尾小节外，再对伴奏进行音高调整，以获得更好听感：低于 G2 的单音高八度，和弦中低于 A2 的音高八度。而比旋律还高的伴奏音符，将会低八度处理。其次，比较旋律和伴奏中起始点接近的音，调整不和谐的大、小二度音及完全相同的音。此外，对于伴奏中重复的相同音，则加上连音符（不包括和弦）。最后，调整旋律和伴奏的力度，以便后期音频制作。

(5) 写入 MIDI 文件 850

完成上述所有步骤后，将旋律和伴奏分别保存为两个 MIDI 文件，便于后期批量渲染音频。

值得注意的是，以上仅为在一些实施例中举例，本申请实施例对此不加以限定。

综上所述，在通过乐曲数据生成乐曲伴奏数据的过程中，通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理，提升对乐曲数据的分析精准性，使得伴奏数据生成过程更加具有针对性；进而综合旋律数据信息、节拍数据信息以及和弦数据信息，在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中，使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现，进而通过乐曲伴奏数据渲染得到能够更精准呈现乐曲数据的乐曲伴奏，提高乐曲伴奏的生成稳定性以及生成效果。

图 10 是本申请一个示例性实施例提供的乐曲伴奏的生成装置的结构框图，如图 10 所示，该装置包括如下部分：

数据获取模块 1010，用于获取乐曲数据，所述乐曲数据被划分为至少两个乐曲节拍，所

述乐曲节拍包括乐曲音符；

信息提取模块 1020，用于从所述乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息，所述节拍数据信息用于描述所述至少两个乐曲节拍的变化速度，所述和弦数据信息用于描述以所述乐曲节拍为单位提取的和弦单元，所述旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况；

数据生成模块 1030，用于根据所述旋律数据信息、所述节拍数据信息以及所述和弦数据信息，生成乐曲伴奏数据，所述旋律数据信息用于以所述乐曲数据的乐曲旋律限定所述乐曲伴奏的伴奏旋律，所述乐曲伴奏数据用于描述所述乐曲数据的伴奏情况；

伴奏生成模块 1040，用于基于乐曲伴奏数据执行音频数据渲染，得到所述乐曲数据对应的所述乐曲伴奏。

在一些实施例中，所述信息提取模块 1020 还用于基于所述至少两个乐曲节拍在单位时间内的数量，确定所述节拍数据信息；提取所述乐曲数据对应的和弦轨和旋律轨，所述和弦轨用于描述所述至少两个乐曲音符以和弦形式构建的音群，所述旋律轨用于描述所述至少两个乐曲音符按照时序关系构成的旋律；基于所述旋律轨和所述和弦轨，确定所述乐曲数据分别对应的所述旋律数据信息和所述和弦数据信息。

在一些实施例中，所述信息提取模块 1020 还用于以所述旋律轨中小节的数量为基准，调整所述和弦轨中小节的数量，得到所述和弦数据信息；基于所述旋律轨中所述至少两个乐曲音符分别对应的音符音高，对所述旋律轨执行旋律变换，得到所述旋律数据信息。

在一些实施例中，所述信息提取模块 1020 还用于获取预设的第一音高和第二音高，所述第一音高和所述第二音高用于限定对所述旋律轨执行旋律变换的变换方式；将所述旋律轨中所述至少两个乐曲音符分别对应的音符音高，分别与所述第一音高和所述第二音高比较，确定第一音符比例和第二音符比例，所述第一音符比例用于表示第一音符占所述至少两个乐曲音符的音符比例，所述第一音符用于表示低于所述第一音高的乐曲音符，所述第二音符比例用于表示第二音符占所述至少两个乐曲音符的音符比例，所述第二音符用于表示高于所述第二音高的乐曲音符；基于所述第一音符比例和所述第二音符比例，对所述旋律轨执行旋律变换，得到所述旋律数据信息。

在一些实施例中，所述数据生成模块 1030 还用于在以乐曲小节为单位的情况下，以所述旋律数据信息为所述乐曲伴奏的生成限制条件，基于所述节拍数据信息和所述和弦数据信息，确定与至少两个乐曲小节分别对应的伴奏织体，一个乐曲小节包括预设数量的乐曲节拍，所述伴奏织体是用于衬托所述旋律数据信息对应的旋律的声音元素；通过所述伴奏织体，调整所述伴奏织体对应的乐曲小节内乐曲音符的音符时值，得到与所述至少两个乐曲小节分别对应的调整小节，所述音符时值用于表示所述乐曲音符的持续时长，所述调整小节包括至少两个调整后的乐曲音符；组合与所述至少两个乐曲小节分别对应的调整小节，生成所述乐曲伴奏数据。

在一些实施例中，所述数据生成模块 1030 还用于获取织体确定规则，所述织体确定规则用于通过乐曲信息提取结果确定与所述乐曲小节对应的所述伴奏织体，所述乐曲信息提取结果包括所述节拍数据信息、所述和弦数据信息以及所述旋律数据信息中的至少之一；在所述织体确定规则的限定下，通过所述旋律数据信息限定所述乐曲伴奏的音符变化幅度，通过所述节拍数据信息限定所述乐曲伴奏的节奏变化情况，通过所述和弦数据信息限定所述乐曲伴奏的音符协调程度，确定与所述至少两个乐曲小节分别对应的所述伴奏织体。

在一些实施例中，所述旋律数据信息用于通过音高范围、音高密度以及音高值，限定所述乐曲伴奏对应的音符变化幅度；

所述节拍数据信息用于通过单位时间内乐曲节拍的数量，确定所述乐曲伴奏对应的节奏类型；

所述和弦数据信息用于通过和弦位置辅助所述旋律数据信息确定音符协调程度。

在一些实施例中,所述伴奏生成模块 1040 还用于对所述旋律数据信息和所述乐曲伴奏数据执行混音处理,得到混音结果;对所述混音结果执行音频数据渲染,得到所述乐曲数据对应的乐曲结果,所述乐曲结果中以所述乐曲伴奏为伴奏内容。

在一些实施例中,所述伴奏生成模块 1040 还用于确定所述旋律数据信息对应的乐曲旋律轨和所述乐曲伴奏数据对应的乐曲伴奏轨;在音频轨对齐的条件下,对所述乐曲旋律轨和所述乐曲伴奏轨执行混音处理,生成所述混音结果。

综上所述,从获取的乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息;以旋律数据信息生成限制条件,通过节拍数据信息和和弦数据信息生成乐曲伴奏数据,最后基于乐曲伴奏数据渲染得到乐曲伴奏。在通过乐曲数据生成乐曲伴奏数据的过程中,通过乐曲信息提取过程对乐曲节拍以及乐曲音符进行量化处理,提升对乐曲数据的分析精准性,使得伴奏数据生成过程更加具有针对性。另外,综合旋律数据信息、节拍数据信息以及和弦数据信息,在乐曲旋律的限定下将节拍和和弦纳入伴奏旋律生成过程中,使得乐曲伴奏数据能够通过音乐参数进行更细致地呈现,使得通过乐曲伴奏数据渲染得到的乐曲数据的乐曲伴奏更加精准,提高乐曲伴奏的生成稳定性以及生成效果。

需要说明的是:上述实施例提供的乐曲伴奏的生成装置,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将设备的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的乐曲伴奏的生成装置与乐曲伴奏的生成方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

图 11 示出了本申请一个示例性实施例提供的计算机设备的结构示意图。该计算机设备 1100 包括中央处理单元(Central Processing Unit, CPU) 1101、包括随机存取存储器(Random Access Memory, RAM) 1102 和只读存储器(Read Only Memory, ROM) 1103 的系统存储器 1104,以及连接系统存储器 1104 和中央处理单元 1101 的系统总线 1105。计算机设备 1100 还包括用于存储操作系统 1113、应用程序 1114 和其他程序模块 1115 的大容量存储设备 1106。

大容量存储设备 1106 通过连接到系统总线 1105 的大容量存储控制器(未示出)连接到中央处理单元 1101。大容量存储设备 1106 及其相关联的计算机可读介质为计算机设备 1100 提供非易失性存储。也就是说,大容量存储设备 1106 可以包括诸如硬盘或者紧凑型光盘只读存储器(Compact Disc Read Only Memory, CD-ROM)驱动器之类的计算机可读介质(未示出)。

不失一般性,计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。上述的系统存储器 1104 和大容量存储设备 1106 可以统称为存储器。

根据本申请的各种实施例,计算机设备 1100 还可以通过诸如因特网等网络连接到网络上的远程计算机运行。也即计算机设备 1100 可以通过连接在系统总线 1105 上的网络接口单元 1111 连接到网络 1112,或者说,也可以使用网络接口单元 1111 来连接到其他类型的网络或远程计算机系统(未示出)。

上述存储器还包括一个或者一个以上的程序,一个或者一个以上程序存储于存储器中,被配置由 CPU 执行。

本申请的实施例还提供了一种计算机设备,该计算机设备包括处理器和存储器,该存储器中存储有至少一段程序,至少一段程序由处理器加载并执行以实现上述各方法实施例提供的乐曲伴奏的生成方法。

本申请的实施例还提供了一种计算机可读存储介质,该计算机可读存储介质上存储有至少一段程序,至少一段程序由处理器加载并执行,以实现上述各方法实施例提供的乐曲伴奏

的生成方法。

本申请的实施例还提供了一种计算机程序产品，该计算机程序产品包括计算机程序，该计算机程序存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机程序，处理器执行该计算机程序，使得该计算机设备执行上述实施例中任一所述的乐曲伴奏的生成方法。

以上所述仅为本申请的可选实施例，并不用以限制本申请，凡在本申请的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本申请的保护范围之内。

权 利 要 求 书

1.一种乐曲伴奏的生成方法，所述方法由计算机设备执行，所述方法包括：

获取乐曲数据，所述乐曲数据被划分为至少两个乐曲节拍，所述乐曲节拍包括乐曲音符；

从所述乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息，所述节拍数据信息用于描述所述至少两个乐曲节拍的变化速度，所述和弦数据信息用于描述以所述乐曲节拍为单位提取的和弦单元，所述旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况；

根据所述旋律数据信息、所述节拍数据信息以及所述和弦数据信息，生成乐曲伴奏数据，所述旋律数据信息用于以所述乐曲数据的乐曲旋律限定所述乐曲伴奏的伴奏旋律，所述乐曲伴奏数据用于描述所述乐曲数据的伴奏情况；

基于所述乐曲伴奏数据执行音频数据渲染，得到所述乐曲数据对应的所述乐曲伴奏。

2.根据权利要求1所述的方法，其中，所述从所述乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息，包括：

基于所述至少两个乐曲节拍在单位时间内的数量，确定所述节拍数据信息；

提取所述乐曲数据对应的和弦轨和旋律轨，所述和弦轨用于描述所述至少两个乐曲音符以和弦形式构建的音群，所述旋律轨用于描述所述至少两个乐曲音符按照时序关系构成的旋律；

基于所述旋律轨和所述和弦轨，确定所述旋律数据信息和所述和弦数据信息。

3.根据权利要求2所述的方法，其中，所述基于所述旋律轨和所述和弦轨，确定所述旋律数据信息和所述和弦数据信息，包括：

以所述旋律轨中小节的数量为基准，调整所述和弦轨中小节的数量，得到所述和弦数据信息；

基于所述旋律轨中所述至少两个乐曲音符分别对应的音符音高，对所述旋律轨执行旋律变换，得到所述旋律数据信息。

4.根据权利要求3所述的方法，其中，所述基于所述旋律轨中所述至少两个乐曲音符分别对应的音符音高，对所述旋律轨执行旋律变换，得到所述旋律数据信息，包括：

获取预设的第一音高和第二音高，所述第一音高和所述第二音高用于限定对所述旋律轨执行旋律变换的变换方式；

将所述旋律轨中所述至少两个乐曲音符分别对应的音符音高，分别与所述第一音高和所述第二音高比较，确定第一音符比例和第二音符比例，所述第一音符比例用于表示第一音符占所述至少两个乐曲音符的音符比例，所述第一音符用于表示低于所述第一音高的乐曲音符，所述第二音符比例用于表示第二音符占所述至少两个乐曲音符的音符比例，所述第二音符用于表示高于所述第二音高的乐曲音符；

基于所述第一音符比例和所述第二音符比例，对所述旋律轨执行旋律变换，得到所述旋律数据信息。

5.根据权利要求1至4任一所述的方法，其中，所述根据所述旋律数据信息、所述节拍数据信息以及所述和弦数据信息，生成乐曲伴奏数据，包括：

在以乐曲小节为单位的情况下，以所述旋律数据信息为所述乐曲伴奏的生成限制条件，基于所述节拍数据信息和所述和弦数据信息，确定与至少两个乐曲小节分别对应的伴奏织体，一个乐曲小节包括预设数量的乐曲节拍，所述伴奏织体是用于衬托所述旋律数据信息对应的

旋律的声音元素；

通过所述伴奏织体，调整所述伴奏织体对应的乐曲小节内乐曲音符的音符时值，得到与所述至少两个乐曲小节分别对应的调整小节，所述音符时值用于表示所述乐曲音符的持续时长，所述调整小节包括至少两个调整后的乐曲音符；

组合与所述至少两个乐曲小节分别对应的调整小节，生成所述乐曲伴奏数据。

6.根据权利要求5所述的方法，其中，所述以所述旋律数据信息为所述乐曲伴奏的生成限制条件，基于所述节拍数据信息和所述和弦数据信息，确定与至少两个乐曲小节分别对应的伴奏织体，包括：

获取织体确定规则，所述织体确定规则用于通过乐曲信息提取结果确定与所述乐曲小节对应的所述伴奏织体，所述乐曲信息提取结果包括所述节拍数据信息、所述和弦数据信息以及所述旋律数据信息中的至少之一；

在所述织体确定规则的限定下，通过所述旋律数据信息限定所述乐曲伴奏的音符变化幅度，通过所述节拍数据信息限定所述乐曲伴奏的节奏变化情况，通过所述和弦数据信息限定所述乐曲伴奏的音符协调程度，确定与所述至少两个乐曲小节分别对应的所述伴奏织体。

7.根据权利要求1至6任一所述的方法，其中，

所述旋律数据信息用于通过音高范围、音高密度以及音高值，限定所述乐曲伴奏对应的音符变化幅度；

所述节拍数据信息用于通过单位时间内乐曲节拍的量，确定所述乐曲伴奏对应的节奏类型；

所述和弦数据信息用于通过和弦位置辅助所述旋律数据信息确定音符协调程度。

8.根据权利要求1至7任一所述的方法，其中，所述方法还包括：

对所述旋律数据信息和所述乐曲伴奏数据执行混音处理，得到混音结果；

对所述混音结果执行音频数据渲染，得到所述乐曲数据对应的乐曲结果，所述乐曲结果以所述乐曲伴奏为伴奏内容。

9.根据权利要求8所述的方法，其中，所述对所述旋律数据信息和所述乐曲伴奏数据执行混音处理，得到混音结果，包括：

确定所述旋律数据信息对应的乐曲旋律轨和所述乐曲伴奏数据对应的乐曲伴奏轨；

在音频轨对齐的条件下，对所述乐曲旋律轨和所述乐曲伴奏轨执行混音处理，生成所述混音结果。

10.一种乐曲伴奏的生成装置，所述装置包括：

数据获取模块，用于获取乐曲数据，所述乐曲数据被划分为至少两个乐曲节拍，所述乐曲节拍包括乐曲音符；

信息提取模块，用于从所述乐曲数据中提取节拍数据信息、和弦数据信息以及旋律数据信息，所述节拍数据信息用于描述所述至少两个乐曲节拍的变化速度，所述和弦数据信息用于描述以所述乐曲节拍为单位提取的和弦单元，所述旋律数据信息用于描述至少两个乐曲音符之间的音符变化情况；

数据生成模块，用于根据所述旋律数据信息、所述节拍数据信息以及所述和弦数据信息，生成乐曲伴奏数据，所述旋律数据信息用于以所述乐曲数据的乐曲旋律限定所述乐曲伴奏的伴奏旋律，所述乐曲伴奏数据用于描述所述乐曲数据的伴奏情况；

伴奏生成模块，用于基于所述乐曲伴奏数据执行音频数据渲染，得到所述乐曲数据对应的所述乐曲伴奏。

11.一种计算机设备，所述计算机设备包括处理器和存储器，所述存储器中存储有至少一段程序，所述至少一段程序由所述处理器加载并执行以实现如权利要求 1 至 9 任一所述的乐曲伴奏的生成方法。

12.一种计算机可读存储介质，所述存储介质中存储有至少一段程序，所述至少一段程序由处理器加载并执行以实现如权利要求 1 至 9 任一所述的乐曲伴奏的生成方法。

13.一种计算机程序产品，包括计算机程序，所述计算机程序被处理器执行时实现如权利要求 1 至 9 任一所述的乐曲伴奏的生成方法。