



(12) 发明专利申请

(10) 申请公布号 CN 101894558 A

(43) 申请公布日 2010. 11. 24

(21) 申请号 201010244883. 2

(22) 申请日 2010. 08. 04

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
总部办公楼

(72) 发明人 张梦晗 李玉龙 张德明 覃景繁

(51) Int. Cl.

G10L 19/00 (2006. 01)

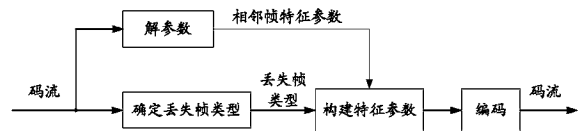
权利要求书 3 页 说明书 17 页 附图 11 页

(54) 发明名称

丢帧恢复方法、设备以及语音增强方法、设备和系统

(57) 摘要

本发明实施例公开了一种丢帧恢复方法、设备及语音增强方法、设备及系统。其中,上述丢帧恢复方法包括:根据丢失帧相邻的相邻帧类型确定丢失帧的类型;获取对相邻帧进行解参数后得到的特征参数,特征参数用于确定帧中编码数据的特征信息;根据相邻帧的特征参数以及各帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数;根据构建的丢失帧的特征参数恢复丢失帧。通过上述方法,可以实现参数域的丢帧恢复,减少恢复过程中消耗的系统资源。



1. 一种丢帧恢复方法,其特征在于,包括如下步骤:
根据与丢失帧相邻的相邻帧类型确定所述丢失帧的类型;
获取对所述相邻帧进行解参数后得到的特征参数,所述特征参数用于确定帧中编码数据的特征信息;
根据相邻帧的特征参数以及所述丢失帧与所述相邻帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数;
根据构建的所述丢失帧的特征参数恢复所述丢失帧。
2. 如权利要求 1 所述的方法,其特征在于,所述根据与丢失帧相邻的相邻帧类型确定丢失帧的类型步骤之前,还包括:
通过检测收到的帧中的帧序号,或者时间戳确定是否符合预设规则的方法来确定所述丢失帧及与所述相邻帧。
3. 如权利要求 1 所述的方法,其特征在于,所述根据与丢失帧相邻的相邻帧类型确定丢失帧的类型包括:
根据所述相邻帧的类型并结合实际帧传输情形,估计所述丢失帧最可能的类型;
和 / 或
根据最佳帧类型确定所述丢失帧的类型。
4. 如权利要求 1 所述的方法,其特征在于,还包括:
结合影响因子,构建与丢失帧类型相符的特征参数。
5. 如权利要求 1 所述的方法,其特征在于,还包括:
当确认有所述丢失帧且需要对所述丢失帧进行恢复时,判断是否进入延时模式,如果是,则使用前后帧的特征参数进行恢复,所述相邻帧包括前后帧;否则,进入非延时模式,使用前帧的特征参数进行恢复,所述相邻帧指前帧。
6. 如权利要求 5 所述的方法,其特征在于,所述判断是否进入延时模式包括:
判断是否存在抖动缓存,如果否,则判断进入非延时模式;
如果是,则判断是否接收到下一帧,如果接收到下一帧,则判断进入延时模式,否则,判断为进入非延时模式。
7. 一种语音增强方法,其特征在于,包括如权利要求 1-6 任一的丢帧恢复方法,用于对语音帧进行丢失帧恢复,所述方法还包括:
对丢失帧恢复后的语音帧进行解参数后得到特征参数,并利用解参数后得到的特征参数进行语音增强处理。
8. 如权利要求 7 所述的方法,其特征在于,所述语音增强处理包括:
在参数域实现自动回声控制、自动噪声抑制、抗削波、自动电平控制以及自动噪声补偿中的一种或多种。
9. 一种丢帧恢复设备,其特征在于,包括:
帧信息确定单元,用于根据与丢失帧相邻的相邻帧类型确定所述丢失帧的类型;
解参数单元,用于获取对所述相邻帧进行解参数后得到的特征参数,所述特征参数用于确定帧中编码数据的特征信息;
特征参数构建单元,用于根据相邻帧的特征参数以及所述丢失帧与所述相邻帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数; ;

恢复单元,用于根据构建的所述丢失帧的特征参数恢复所述丢失帧。

10. 如权利要求 9 所述的设备,其特征在于,所述帧信息确定单元包括:

丢失帧确定单元,用于通过检测收到的帧中的帧序号,或者时间戳确定是否符合预设规则的方法来确定所述丢失帧及所述相邻帧;

帧类型确定单元,用于通过帧中相关标志位确定帧的类型;

丢失帧类型确定单元,用于根据所述帧类型确定单元确定的相邻帧类型确定所述丢失帧类型。

11. 如权利要求 9 所述的设备,其特征在于,还包括:

模式选择单元,用于当丢失帧确定单元确定有丢失帧且需要恢复时,判断是否进入延时模式,如果是,则所述特征参数构建单元使用前后帧的特征参数进行恢复,所述相邻帧包括前后帧;否则,进入非延时模式,所述特征参数构建单元使用前帧的特征参数进行恢复,所述相邻帧指前后帧。

12. 一种语音增强设备,其特征在于,包括如权利要求 9-11 任一的丢帧恢复设备,还包括以下几个语音增强单元中的一个或多个:

自动回声控制单元,用于自动回声控制;

自动噪声抑制单元,用于自动噪声抑制;

抗削波单元,用于抗削波;

自动电平控制单元,用于自动电平控制;

自动噪声补偿单元,用于自动噪声补偿;

上述一个或多个单元对丢失帧恢复后的语音帧进行解参数后得到特征参数,并利用解参数后得到的特征参数进行语音增强处理。

13. 如权利要求 12 所述的设备,其特征在于,当包括上述多个语音增强单元时,还包括:

分类单元,用于对输入的信号进行分类,确定信号的类型,并根据确定的类型送至自动回声控制单元、自动噪声抑制单元、抗削波单元、自动电平控制单元、自动电平控制单元以及自动噪声补偿单元中的一个或多个进行处理。

14. 一种语音增强系统,其特征在于,包括:

第一基站,第一终端;第二基站,第二终端;权利要求 9-11 中任一所述的丢帧恢复设备;

其中,所述第一基站与所述第一终端之间通过第一链路进行连接;所述第二基站与所述第二终端之间通过第二链路进行连接;所述第一基站与所述第二基站之间通过第三链路进行连接;

丢帧恢复设备用于对所述第一链路、所述第二链路或者所述第三链路中的一个或多个环节出现的丢帧进行恢复。

15. 一种语音增强系统,其特征在于,包括:

第一基站,第一终端;第二基站,第二终端;权利要求 12-13 任一的语音增强设备;

其中,第一基站与第一终端之间通过第一链路进行连接;第二基站与第二终端之间通过第二链路进行连接;第一基站与第二基站之间通过第三链路进行连接;

语音增强设备用于对所述第一链路、所述第二链路或者所述第三链路中的一个或多个

环节出现的丢帧进行恢复及实现上述一个或多个环节出现的语音增强。

丢帧恢复方法、设备以及语音增强方法、设备和系统

技术领域

[0001] 本发明涉及通信技术领域,尤其涉及一种丢帧恢复方法、设备以及一种语音增强方法、设备和系统。

背景技术

[0002] 语音传输在通信领域有着广泛的应用,一个常见的语音传输系统如图 1 所示:原始语音数据(模拟信号)经过采样得到基于样点值的波形信号(数字信号),再经过编码后形成以帧为基本单元的码流,并通过相应的传输网络(如 IP 网、TFO 网)进行传输。由于受各种网络环境因素的制约,会出现网络拥塞以及抖动等现象,导致帧丢失,从而对语音质量造成影响。

[0003] 为了实现语音增强(即提高语音质量),现有技术一般都采用线性域(波形域)补偿的方法来对丢失的帧进行恢复,即先将编码后的相关帧进行完全解码,得到基于样点值的信号波形,再根据前一帧或后一帧的波形来恢复当前帧的波形,将恢复出来的波形(基于样点值)进行编码,恢复生成丢失的帧数据。

[0004] 发明人在实现本发明的过程中,发现现有技术至少存在如下缺点:

[0005] 由于现有技术在进行帧恢复时,需要先对相关帧进行解码,然后再根据解码后的相关帧的波形进行恢复;而解码过程往往需要消耗较多的系统资源(如内存占用、CPU 占用率等),因此,如果当网络条件较差,需要恢复的帧数据较多时,需要执行多次解码操作,从而大大增加了系统资源消耗。

发明内容

[0006] 本发明实施例提供一种丢帧恢复方法、设备及语音增强方法、设备及系统,用于降低丢帧恢复过程中消耗的系统资源以及实现语音增强。

[0007] 其中,本发明实施例提供了一种丢帧恢复方法,包括如下步骤:

[0008] 根据与丢失帧相邻的相邻帧类型确定丢失帧的类型;

[0009] 获取对相邻帧进行解参数后得到的特征参数,特征参数用于确定帧中编码数据的特征信息;

[0010] 根据相邻帧的特征参数以及所述丢失帧与所述相邻帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数;

[0011] 根据构建的丢失帧的特征参数恢复丢失帧。

[0012] 本发明实施例还提供了一种语音增强方法,包括上述丢帧恢复方法,用于对语音帧进行丢失帧恢复,还包括:

[0013] 在对丢失帧进行恢复后,通过在参数域实现自动回声控制、自动噪声抑制、抗削波、自动电平控制以及自动噪声补偿中的一种或多种技术对进行丢失帧恢复后的语音帧信号进行语音增强。

[0014] 本发明实施例还提供了一种丢帧恢复设备,包括:

- [0015] 帧信息确定单元,用于根据与丢失帧相邻的相邻帧类型确定丢失帧的类型;
- [0016] 解参数单元,用于获取对相邻帧进行解参数后得到的特征参数,特征参数用于确定帧中编码数据的特征信息;
- [0017] 特征参数构建单元,用于根据相邻帧的特征参数以及所述丢失帧与所述相邻帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数;
- [0018] 恢复单元,用于根据构建的丢失帧的特征参数恢复丢失帧。
- [0019] 本发明实施例还包括一种语音增强设备,包括以上丢帧恢复设备,还包括以下几个语音增强单元中的一个或多个: :
- [0020] 自动回声控制单元,用于自动回声控制;
- [0021] 自动噪声抑制单元,用于自动噪声抑制;
- [0022] 抗削波单元,用于抗削波;
- [0023] 自动电平控制单元,用于自动电平控制;
- [0024] 自动噪声补偿单元,用于自动噪声补偿;
- [0025] 上述一个或多个语音增强单元对丢帧恢复设备进行丢失帧恢复后的语音帧进行解参数后得到特征参数,并利用解参数后得到的特征参数进行语音增强处理。
- [0026] 本发明实施例还提供了一种语音增强系统,包括:
- [0027] 第一基站,第一终端;第二基站,第二终端;上述丢帧恢复设备;
- [0028] 其中,第一基站与第一终端之间通过第一链路进行连接;第二基站与第二终端之间通过第二链路进行连接;第一基站与第二基站之间通过第三链路进行连接;
- [0029] 丢帧恢复设备用于对第一链路、第二链路或者第三链路中的一个或多个环节出现的丢帧进行恢复。
- [0030] 本发明实施例还提供了一种语音增强系统,包括:
- [0031] 第一基站,第一终端;第二基站,第二终端;上述语音增强设备;
- [0032] 其中,第一基站与第一终端之间通过第一链路进行连接;第二基站与第二终端之间通过第二链路进行连接;第一基站与第二基站之间通过第三链路进行连接;
- [0033] 语音增强设备用于对第一链路、第二链路或者第三链路中的一个或多个环节出现的丢帧进行恢复及实现上述一个或多个环节出现的语音增强。
- [0034] 上述技术方案中具有如下的优点:
- [0035] 本发明实施例中丢帧恢复方法及设备中,通过利用相邻帧的类型与特征参数来恢复丢失帧,由于解参数过程相比于完全解码到波形域的过程所消耗的资源要少得多,因此,通过本发明实施例恢复方法可以大大降低系统消耗的资源,在网络条件较差,存在很多丢失帧的情况下,这种效果更加明显。
- [0036] 同时,本发明实施例语音增强方法、设备及系统中,在基于丢帧恢复的基础上进行对帧进行解参数,并对解参数后得到的特征参数进行语音增强,可以在降低系统消耗的资源上更进一步提升语音的质量。

附图说明

- [0037] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些

实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

- [0038] 图 1 为现有技术常见语音传输系统示意图;
- [0039] 图 2 为本发明实施例一方法流程示意图;
- [0040] 图 3 为本发明实施例一方法流程图;
- [0041] 图 4 为本发明实施例二加入模式选择的方法流程图;
- [0042] 图 5 为本发明实施例三采用 AMR-NB 进行语音增强示意图;
- [0043] 图 6 为本发明实施例三采用 AMR-NB 对丢失帧特征参数进行恢复的方法;
- [0044] 图 7 为本发明实施例四在基于丢失帧恢复基础上通过其他参数域进行语音增强的方法示意图;
- [0045] 图 8 为本发明实施例四中加入信号分类器对信号进行分类示意图;
- [0046] 图 9 为本发明实施例在参数域进行抗削波方法流程图;
- [0047] 图 10 为本发明实施例五丢帧恢复设备结构示意图;
- [0048] 图 11 为本发明实施例五丢帧恢复设备中帧信息确定单元结构示意图;
- [0049] 图 12 为本发明实施例五丢帧恢复设备包括模式选择单元时的结构示意图;
- [0050] 图 13 为本发明实施例五语音增强设备结构示意图;
- [0051] 图 14 为本发明实施例五语音增强设备一个具体硬件实现示意图;
- [0052] 图 15 为本发明实施例五语音增强设备另一个具体硬件实现示意图;
- [0053] 图 16 为本发明实施例六应用语音增强设备系统的示意图;
- [0054] 图 17 为本发明实施例六在全 IP 组网下 TrFO 场景下应用示意图;
- [0055] 图 18 为本发明实施例六在 TFO 场景下应用示意图;
- [0056] 图 19 为本发明实施例六在 TDM 场景下应用示意图。

具体实施方式

[0057] 为使本发明的目的、技术方案及优点更加清楚明白,以下将通过具体实施例和相关附图,对本发明作进一步详细说明。

[0058] 实施例一

[0059] 本发明实施例一提供了一种语音增强方法,参见图 2,为本发明实施例一个流程示意图,在接收码流后,首先根据相邻帧的类型确定该丢失帧的类型,再利用对相邻帧进行解参数后得到的特征参数来构建丢失帧的特征参数,最后对构建的丢失帧的特征参数进行编码,恢复丢失帧。由于本发明实施例是通过解参数后通过得到特征参数来恢复丢失帧,因此,可以看成是一个在“参数域”进行恢复的过程。具体的,参见图 3,本发明实施例包括如下步骤:

[0060] S101、根据与丢失帧相邻的相邻帧类型确定丢失帧的类型;

[0061] 在确定根据相邻的帧的类型确定丢失帧类型之前,首先要确定丢失帧是哪一帧,以及丢失帧相邻的帧,以及这些相邻帧的类型。

[0062] 其中,确定丢失帧以与其相邻的帧可以通过检测收到的帧中的帧序号,或者时间戳是否符合预设规则等方法来确定。

[0063] 具体的,通过帧序号实现的方法是将每个帧按一定规则(如顺序递增)编上序号,

如果接收到发现收到的多个帧序号不满足该规则（如顺序递增过程中中间少了个号），则认为发生了丢帧，那么，该帧的前一帧或前几帧，后一帧或后几帧为该丢失帧的相邻帧。

[0064] 上述收到的多个帧序号可以通过在收到帧数据时先对帧进行缓存，判断是否丢失帧时，解析缓存的帧数据得到；或者也可以在收到帧数据时先解析得到帧序号，将其缓存，在判断时直接读取缓存的帧序号时得到。后续其他与帧相关的数据（如时间戳、帧类型标志、特征参数等）都可以采用先缓存帧，使用时解析帧提取相关数据；或者直接在收到帧时先解析出这些数据进行缓存，需要用到时直接使用这些缓存的数据。

[0065] 具体的，当采用时间戳实现时，每个帧在发送时会打上时间戳，接收端收到多个帧后，根据这些时间戳的先后顺序来判断是否发生丢帧。例如，假设正常时每隔 1ms 就会打上一个时间戳，接收端收到两个帧后，发现两帧间隔了 2ms 或 2ms 以上，则认为在这两个帧之间有帧丢失，这两个帧可认为是相邻帧，可以理解的是，前一相邻帧再往前的一帧或多帧以及再往后一相邻帧再往后一帧的一帧或多帧也可认为是相邻帧。

[0066] 此外，在某一些情况下，如果通过检测发现收到的为错帧，则也可将这种情况认为该帧为丢失帧；当然，实际应用中，也可以不认为是丢失帧，而是按错帧的情况进行处理。

[0067] 需要说明的是，上述丢失帧可以是一个或者多个，如果是多个，则恢复时先恢复第一个，然后再依次利用恢复的帧（也可以是一个或多个）来恢复后面的丢失帧。

[0068] 在确定了帧丢失的情况，以及丢失帧的相邻帧后，便可以通过相邻帧的类型来确定丢失帧的类型。

[0069] 其中，在确定帧的类型时，可以通过帧中的相应标志位来确定。例如，基于 AMR-NB 编码方式的帧在编码过程中，编码器在将语音数据编码成帧数据时，会在帧特定的位置通过标志位来表示帧的类型（如话音帧、舒适噪声帧、空帧等）；接收端只要接收到帧数据，便可以根据相应的标志位来确定帧的类型；基于 EVRC(Enhanced Variable Rate Codec, 增强型变速率语音编解码) 的帧通过速率标志位以速率为类型区分不同类型的帧（如 1/8 速率，半速率，全速率等）。

[0070] 通过以上方法确定丢失帧的相邻帧后，可以根据这些相邻帧的类型来确定丢失帧的类型。具体的，可以根据相邻帧的类型并结合实际帧传输情形，估计丢失帧最可能的类型，例如，假设码流中的帧类型包括 A、B、C 三种，实际应用中，大多数情况下这几种类型的帧会按 A、B、C 三种类型先后顺序传输，此时，如果丢失帧的前一帧为 A、后一帧为 C，那么丢失帧为 B 的概率为最大，可以在这种情况下认为该丢失帧的类型为 B；或者，再结合根据最佳帧类型确定丢失帧的类型，例如，针对 B 帧，可能是好帧（没有损失的数据），也可能是坏帧（如某一位或几位出错，但有一部分也能用），这种情况下，都按是好帧（即最佳帧类型）的情况进行恢复。实际使用过程中，一般都会使用上述两种方案的结合，当然，也可以只基于其中一种方法进行恢复，本发明实施例并不进行限定。

[0071] S102、获取对相邻帧进行解参数后得到的特征参数，特征参数用于确定帧中编码数据的特征信息；

[0072] 在进行解参数的过程中，由于进行编解码时基于的语音数字信号的模型可能不同，因此，进行解参数后得到的特征参数也会不同。例如，以基于 CELP(Code Excited Linear Prediction, 码激励线性预测编码) 模型的解码器为例，输出的特征参数可以包括：线谱频率 (LSF, Line Spectrum Frequency) 或线谱对 (LSP, Line Spectrum Pair)、基音延

迟、自适应码书增益 (Gain Pit)、固定码书增益 (Gain Code)、反射系数等信息,其中,线谱频率以及线谱对是用于描述同一信息采用的两种不同表示方法,两者之间可以相互转化。以基于 VSELP (Vector Sum Excited Linear Prediction, 矢量和激励线性预测编码) 模型为例,解参数后可以得到谱包络、帧能量等参数。上述特征参数的具体定义以及解参数的方法都已在相应的标准中有定义,本领域技术人员可以根据标准中的定义完成解参数的步骤,在此不再赘述。

[0073] 其中,上述各个特征参数用于确定帧中编码数据的特征信息;这里的编码数据即对语音信号进行采样、编码后得到的数据;编码数据中的特征信息用于描述编码信号所对应的时域或频域的一些信息,而解参数后得到的特征参数即用于对这些特征信息进行描述或表示。例如,LSF/LSP 描述了信号在频域上的包络特性;基音延迟描述了当前帧的基音与前一帧的差值;自适应码书增益描述了频域周期性最强的频谱分量的能量特性;固定码书增益描述了其他周期性较弱的频谱分量的能量特性。如果一个帧的特征参数确定了,那么就可以根据这些特征参数来对该帧进行恢复。

[0074] 需要说明的是,当本发明实施例基于一个独立的设备时,上述解参数过程的由该独立设备的解参数单元来完成。而如果本发明实施例基于一个现有的设备,那么解参数功能可以由现有设备来完成,例如,现有设备为了转码或兼容需求,已有一个解码器,而该解码器在解码(解码到波形)过程中,也会输出特征参数,那么,就可以利用该解码器输出的特征参数;当然,这种情况下也可以不使用原有设备输出的特征参数,而是在本设备上通过一个解参数单元来完成特征参数的输出。

[0075] S103、根据相邻帧的特征参数以及所述丢失帧与相邻帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数;

[0076] 确定丢失帧的类型后,需要构建与丢失帧类型相符的特征参数,例如,对于一些类型的丢失帧(话音帧),恢复时只需要使用相邻帧中解参数出来的一种特征参数;而对于另一些类型的丢失帧(如信号帧),需要使用多种特征参数。因此,根据不同类型的帧,需要构建与其类型相符的特征参数,这样才能根据这些构建的特征参数对丢失帧进行恢复。

[0077] 需要说明的是,对于另一些类型的丢失帧(如静音帧等),可以不进行恢复,直接透传到后级进行处理(如后级进行一些数据的填充),具体处理方法在本发明实施例中不进行详细描述。

[0078] 构建与丢失帧类型相符的特征参数根据相邻帧的特征参数以及各帧中编码数据特征信息之间的相关性来完成。这里的相关性也可以理解为表征各帧中编码数据的特征信息变化趋势。实际应用中,各帧中的编码数据的特征信息并非完全孤立的,很多情况下都是具有相关性的,如多个帧中编码数据的包络特性(通过 LSF/LSP 特征参数体现)会表现出增加,或者减少,或者有规律变化的趋势等,通过这种相关性(或变化趋势),可以利用某一些帧来对另一些帧进行恢复。本发明实施例中,丢失帧中编码数据特征信息与相邻帧中的编码数据特征信息也会存在相关性,因此,可以通过相邻帧的特征参数(用于表征相邻帧中的特征信息)以及这种相关性来得到丢失帧的特征参数,例如,前 2 帧相邻帧特征参数(也可理解为特征信息,在此并不严格区分)为分别为 2、4;后 2 帧分别为 7、10,则可以看到各帧的相关性(变化趋势)为呈现增加的趋势,因此,可得到丢失帧在很大概率上为 4-7 之间的一个值。

[0079] 在根据多个帧中编码数据特征信息之间的相关性进行丢失帧特征参数恢复时,选取的相邻帧数越多,判断这种相关性(或变化趋势)的准确度也越高,但同时引入的时延以及算法复杂度也会增加,实际使用中可以根据应用需求选择合适数量的相邻帧(如前2帧,或前后各一帧)来恢复丢失帧。

[0080] 恢复丢失帧所采用的算法的实现形式并不唯一,例如,想恢复LSF/LSF参数时,一个简单的算法是将丢失帧的前后各一帧的LSF/LSF参数相加后取算术平均,来得到当前帧的LSF/LSF参数;或者,想恢复基音延迟参数时,将前一帧的基音延迟参数加上一固定值或者直接使用前一帧的基音延迟参数。

[0081] 实际应用当中,为了取得更好性能的恢复效果,可以在利用丢失帧与相邻帧之间的相关性的基础上结合引入影响因子(或加权系数)等方式创建更优的算法,实现对丢失帧特征参数更准确的构建,例如,虽然有多个帧会与丢失帧相关,但因为相关性不同,所以需要相关的帧的相关程度通过影响因子进行限定(加权)。影响因子的确定可以通过历史经验值或者测试值来得到,通过结合影响因子,可以取得更好的恢复效果。

[0082] S104、根据构建的特征参数恢复丢失帧

[0083] 构建完丢失帧的特征参数之后,便可以根据这些构建的特征参数恢复丢失帧,其具体实现也可参见标准相关定义,将这些特征参数编码后恢复成新的帧。同时,与解参数过程类型,根据本发明实施例基于的设备可以由一个独立设备中的相关功能单元来完成,也可以基于原有设备中具有相应功能的单元来完成,在此并不限定。

[0084] 本发明实施例中,通过利用相邻帧的类型与特征参数来恢复丢失帧,由于解参数过程相比于完全解码到波形域的过程所消耗的资源要少得多,因此,通过本发明实施例恢复方法可以大大降低系统消耗的资源,在网络条件较差,存在很多丢失帧的情况下,这种效果更加明显。

[0085] 实施例二

[0086] 本发明实施例基于上述实施例一,提供了一种基于不同工作模式的语音增强方法,为了应对实际应用中传输质量与延迟的要求,在确定有丢失帧且需要对丢失帧进行恢复时,通过进入不同的模式来应对不同的实际应用场景。

[0087] 这里的系统指的是应用了本发明实施例语音增强技术的系统;在该系统中,为了防止网络抖动(Jitter),可能会设立抖动缓存(Jitter Buffer),通过缓存一定量的帧,并进行重排序来防止抖动,通过抖动缓存的设置,使得传输更加稳定,提高了传输质量,但在另一方面却会造成延时。

[0088] 当系统提供了Jitter Buffer时,在获取跟帧有关的数据(如序号、时间戳、类型等)时也可以从Jitter Buffer中去取(在这种情况下,Jitter Buffer需要支持提供某一接口,如标志位、接口函数等来供其他单元获取相关的信息)。通过利用系统提供的Jitter Buffer,可以不必再开辟一块缓存区缓存相关的数据,节省了存储空间,实现起来也更简单。当然,本发明实施例也可以不使用系统提供的Jitter Buffer,而再设计一个缓存(类似于实施例一中的用于缓存帧的缓存)来实现相应的功能。

[0089] 由于系统的实际应用场景并不固定,不同的场景对传输质量与传输延时要求并不相同,有的应用场景对传输延时的要求比较高,有的则对传输质量要求比较高,因此,本发明实施例引入了两种工作模式来针对实际中可能会遇到的两种场景来满足可能会出现

场景,具体包括:延时模式(Delay Mode)以及非延时模式(Non-Delay Mode),当进入延时模式时,用前一帧(或前几帧)和后一帧(后几帧)进行恢复,以保证对质量的要求;当进入非延时模式时,只用前一帧(或前几帧)进行恢复,以保证对时延的要求。

[0090] 具体的,参见图4;包括如下步骤:

[0091] S201、判断系统是否进入延时模式,如果是,执行步骤S202;否执,执行步骤S203;

[0092] 其中,是否进入延时模式可以根据是否有抖动缓存(或者抖动缓存是否工作),以及是否接收到下一帧来判断。具体的,包括如下步骤:

[0093] S211、判断是否存在抖动缓存(Jitter Buffer);如果是,执行步骤S212,否则,判断为进入非延迟模式执行步骤S203;

[0094] S212、判断是否接收到下一帧;如果是,执行步骤S202,否则,判断为进入非延迟模式,执行步骤S203;

[0095] 如果在准备发送最近接收到的帧数据时,已经收到下一帧,则判定进行入延迟模式,利用前后帧的特征参数进行恢复;否则,为了减少延时,进入非延迟模式,仅使用前帧(前一帧或前几帧)的特征参数进行恢复。

[0096] S202、进入延迟模式,利用前后帧的特征参数进行恢复;

[0097] 在延迟模式下,可以利用前后帧的特征参数进行恢复,此时,相邻帧包括前后帧。本发明例中,仅使用前后各一帧的特征参数进行恢复,但实际使用中,也可以使用前一帧或多帧以及后一帧或多帧的特征参数来恢复。由于用到了前后相邻帧的数据,因此,准确性相对较高,满足了对质量的要求。

[0098] S203、进入非延迟模式,利用前帧的特征参数进行恢复;

[0099] 在非延迟模式下,对延时的要求比较严格,因此,仅使用前一帧(或前几帧)来恢复,此时,相邻帧指的是前帧。由于只使用前一帧(或前几帧),相比于S202中的方案,需要缓存的帧也会减少(不需要缓存后几帧),因此,引入的时延也会减少;同时,少了后一帧或后几帧的特征参数作为参考,在算法实现上也会更加快捷,从而能更快地进行恢复,但由于少了后一帧或后几帧的特征参数作为参考,其准确性可能会不如延迟模式下得到的结果。

[0100] 实际使用过程中,除了上述判断方法外,也可以通过其他类似的方法来选择进入的模式,在此并不限定。例如:可以通过手动配套结合自动检测的方法来进入相应的模式,具体的,用户可以通过相应的程序接口根据实际需求配置某个参数来决定要进入的模式,系统运行时读取该参数来决定选择进入哪种模式;

[0101] 或者,结合上述两种方法,系统通过步骤S201的方法以及读取用户手动配置的参数,判断是否一致,如果一致,则进入相应的模式;如果不一致,则根据一定的算法(如通过判断两种方法的优先级)选择其中一个模式。

[0102] 本发明实施例通过设定两种模式(延迟模式以及非延迟模式),在系统运行时,可以根据实际情况自动选择或者通过人工手动配置进入哪种模式,满足了不同场景下对传输质量及传输时延的要求。

[0103] 实施例三

[0104] 本发明实施例基于上述实施例,以系统采用AMR-NB(Adaptive Multi-Rate Narrow-Band,自适应多码率-窄带)编码为例对语音增强技术进行详细说明,参见图5,包括如下步骤:

[0105] S301、获取帧数据；

[0106] 获取帧数据可以是直接获取前端处理后的帧数据，或者在本端增加一个接收单元通过解析网络传输的数据包 (Packet) 得到。在实际应用中，为了适应各种网络传输协议，编码后的帧都会根据网络传输类型进行打包（如打成 IP 包、RTP 包等），每个包一般包括只包含一个帧，这样即使发生丢包的情形，损失的帧也比较少，可以提高传输的可靠性；在一些需要传输多路数据的应用场景下，一个包也可能包含多个帧，但多个帧分别属于多个用户，即一个用户对应于包中的一帧数据，这样也是为了在发生丢包情形时不至于损失过多的帧数据。

[0107] 在本发明实施例中，为了说明方便，以一个数据包只含有一帧为例来进行说明，在这种情况下，无论是“丢包”还是“丢帧”，都意味着那一帧语音数据的丢失，只不过“丢包”是从协议传输层面上来讲，而“丢帧”是从数据层面上来讲。因此，本发明实施例并不对“丢包”以及“丢帧”作严格区分，可以认为“丢包”即意味着出现了“丢帧”，而“丢帧”则意味着是因为“丢包”进而导致了“丢帧”。

[0108] 需要指出的是，如果一个包中包含有多个属于同一用户的帧时，只需通过解析出包中的多个帧数据即可，本发明实施例中其余步骤也完全对这种情况适用，后续恢复步骤都可以参考一个包只包含一个帧情况的步骤进行。此外，如果一帧数据由多个包构成，则只需要解析多个包后得到该帧数据，后续步骤也可以参考一个包只包含一个帧情况的步骤进行。

[0109] 本发明实施例中，解析包的过程由具体的传输协议进行定义，解析后得到的包中的净荷 (Payload) 部分一般就是有效的帧数据。

[0110] S302、判断是否有帧丢失，如果是，执行步骤 S303，否则，执行步骤 S308、解参数后得到特征参数，更新缓存的特征参数；后续执行步骤 S307、按正常帧的处理流程来进行处理。

[0111] 其中，步骤 S308 中，用于对帧进行解参数，得到特征参数，并更新缓存的特征参数。这里“缓存的特征参数”是指在系统开辟的一块缓存区域中存储的之前帧的特征参数；因为构建当前帧的特征参数需要用到相邻帧的特征参数，因此，需要将这些相邻帧进行解参数后进行缓存。缓存的大小可以视恢复时使用到的帧数而定，例如，在恢复丢失帧时，需要用到丢失帧前面 3 帧的特征参数来构建丢失帧的特征参数，则可以缓存 3 帧或 3 帧以上（具体可视实际用而定）的特征参数。当本次有新数据时，更新缓存区，丢弃掉最老的数据，得到最近的一帧或几帧数据的特征参数。

[0112] 需要说明的是，本发明实施例中，缓存的数据是解参数后得到的特征参数，实现使用中，也可以缓存帧数据，在需要进行帧恢复时，取出缓存的帧数据，再进行解参数，得到相应的特征参数。这两种方法都有各自的特点，当采用缓存特征参数时，在每一次收到正常帧时都需要执行解参数的步骤，会增加平均资源消耗，但在对丢失帧进行恢复时，可以直接使用缓存的特征参数，不再执行解参数的动作，此时就会降低峰值的资源消耗；而采用缓存帧的方法时，在收到正常帧时并不需要执行解参数，会降低平均资源消耗，但在恢复过程中，需要进行解参数，并执行恢复算法，会增加峰值的资源消耗。在系统设计过程当中，设计指标一般都会以峰值的资源消耗作为参考，如系统总的资源为 12，每个用户在峰值时消耗为 4，则可以设计 3 路用户，如果用户峰值资源消耗为 2，则可以设计 6 路用户。而本发明实施

例中,使用缓存解参数的方法,可以使得在恢复过程中不需要再进行解参数,从而降低系统在峰值的资源消耗,以便更好地满足系统设计需求。

[0113] 判断的具体方法可以参见实施例一中的相关描述,例如,通过序号、时间戳等方法来进行判断。

[0114] S302'、确定进入的模式;具体的方法可以参见上述实施例二,在此不再赘述;该步骤为可选步骤,实际应用中也可以不进行判断,而直接进入某一种模式,使用前一帧(前几帧)和/或后一帧(后几帧)的特征参数来恢复。

[0115] S303、确定丢失帧的类型;

[0116] 具体的,通过与丢失帧相邻的帧(相邻帧)来确定丢失帧的类型。

[0117] 参见表1,本发明实施例基于实施一步骤S101中介绍的帧类型确定方法给出了一种基于AMR-NB编码的帧类型确定方法。

[0118] 表1AMR-NB帧类型确定对应表

[0119]

| 工作模式 | 前一帧 | 后一帧 | 丢失帧 |
|------------|---------------|---------------|----------------|
| 非延时模式 | RX_SPEECH_xxx | | RX_SPEECH_GOOD |
| | RX_SID_xxx | | RX_NO_DATA |
| | RX_NO_DATA | | RX_NO_DATA |
| 延时模式 | RX_SPEECH_xxx | RX_SPEECH_xxx | RX_SPEECH_GOOD |
| | RX_SPEECH_xxx | RX_SID_xxx | RX_SPEECH_GOOD |
| | RX_SPEECH_xxx | RX_NO_DATA | RX_SID_FIRST |
| | RX_SID_xxx | RX_SPEECH_xxx | RX_SPEECH_GOOD |
| | RX_SID_xxx | RX_SID_xxx | RX_NO_DATA |
| | RX_SID_xxx | RX_NO_DATA | RX_NO_DATA |
| | RX_NO_DATA | RX_SPEECH_xxx | RX_NO_DATA |
| | RX_NO_DATA | RX_SID_xxx | RX_NO_DATA |
| RX_NO_DATA | RX_NO_DATA | RX_NO_DATA | |

[0120] 表1中,第一列为系统所处的工作模式,即对应实施例二中的非延时模式以及延时模式;第二、第三列为在各种模式下丢失帧前一帧以及后一帧的类型,采用非延时模式时,后一帧类型为空;最后一列为根据前一帧,或者根据前一帧以及后一帧后确定的丢失帧类型。

[0121] 具体的,以RX_SPEECH打头的为话音帧,包括RX_SPEECH_GOOD、RX_SPEECH_DEGRADED及RX_SPEECH_BAD等类型,表中的RX_SPEECH_xxx包括上述三种类型;

[0122] 以RX_SID打头的为信号帧,包括RX_SID_FIRST、RX_SID_UPDATE及RX_SID_BAD等类型,表中RX_SID_xxx包括上述三种类型;

[0123] 此外,RX_NO_DATA表示静音帧;

[0124] 通过本发明实施例中的表格中的对应关系,可以根据前一帧(或结合后一帧)的类型来决定丢失帧的类型,例如,在非延时模式下,如果前一帧类型为RX_SPEECH_xxx类型,则丢失帧类型为RX_SPEECH_GOOD(尽量以好帧类型去恢复,以得到更好的效果);在延时模式下,如果前一帧类型为RX_SPEECH_xxx,后一帧类型为RX_SID_xxx,则丢失帧类型为RX_SPEECH_GOOD。

[0125] 需要说明的是,上述表格只是本实施例中的一个具体实现形式,具体的需要用到的相邻帧个数(如不采用一帧数据,而采用前几帧和/或后几帧)以及具体定义方法(如前一帧为 RX_SPEECH_DEGRADED 时,丢失帧不为 RX_SPEECH_GOOD,而是 RX_SPEECH_DEGRADED)都不限定。

[0126] 通过对帧类型的确定,后续可以对类型为话音帧的丢失帧进行恢复;而对于信号帧以及静音帧,本发明实施例可以不需要通过恢复算法进行恢复(即可认为没发生丢帧),后续由其他后级单元通过噪声补偿等方法来实现数据恢复,在此并不赘述。

[0127] S304、获取相邻帧的特征参数;

[0128] 通过对相关的相邻帧进行解参数,获取相邻帧的特征参数;解参数的具体方法可以通过 AMR-NB 解码器来完成,AMR-NB 解码器基于 CELP 模型,生成的特征参数主要包括 LSF/LSP、基音延迟、自适应码书增益、固定码书增益等。

[0129] S305、构建与丢失帧类型相符的特征参数;

[0130] 即通过根据相邻帧的特征参数以及多个帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数。

[0131] 例如,以构建丢失帧的 LSF 参数为例,假设丢失帧的 LSF 参数为 LSF(n),选取该帧前 N 帧以及后一帧相邻帧的 LSF 参数进行构建,则前 N 帧及后一帧的 LSF 参数分别为 LSF(n-N)、LSF(n-N+1)...LSF(n-1) 以及 LSF(n+1),总共 N+1 个值;这里的 n 可以为整数,N 可以为正整数;则可以通过如下公式进行恢复:

$$[0132] \quad LSF(n) = mean_lsf + \sum_{i=1}^N fac(i) * LSF_r(n-i) + fac(N+1) * LSF_r(n+1) \quad (\text{公式 1})$$

[0133] 其中,公式中的 mean_lsf 表示经过大量语音数据训练得到的 LSF 平均值(即取大量数据的 LSF 参数的平均值);fac(n) 为影响因子(也可认为是“加权系数”),其值也由大量语音数据训练得到,例如,想确定 fac(1) 时,可以先取一段大量的语音数据,如 1000 帧,然后,假设第 1 帧丢失,此时,先取 fac(1) 为其中一个数,利用公式恢复出第 1 帧的 LSF 参数后,与真实的第 1 帧 LSF 参数值做差,此时,一般差值都会较大;然后,再继续调整 fac(1) 为另一个数,再算 LSF 参数,再与真实值做差,如此反复,直到通过公式逼近的第 1 帧 LSF 参数与真实的 LSF 参数差在一个阈值范围内,则此时 fac(1) 即为训练后得到的 fac(1) 值。同理,其余影响因子也可以通过类似方法进行确定。

[0134] 需要说明的是,上述公式及各参数及实现形式并不唯一,例如,可以取不同数量的相邻帧,或者使用其他加权的方法,只要能利用各帧相关性来构建丢失帧的算法都可以。

[0135] 再次,以需要构建出丢失帧的基音延迟参数时,首先需要得到前后帧的基音周期,然后再生成基音延迟,其中,如果根据基音周期生成基音延迟的方法为本领域技术人员熟知的技术,在此不再赘述。

[0136] 在获取基音周期时,由于语音的基音周期具有较好的平稳性,因此,可以利用丢失帧前几帧和后几帧的变化趋势来判断丢失帧基音的几种变化趋势。这里以丢失帧前两帧和后一帧为例来判断丢失帧基音的 5 种变化趋势(快速上升,快速下降,平稳上升,平稳下降,持平)。

[0137] 假设丢失帧的基音周期为 Delay(n), (n 为整数),丢失帧前两帧的基音周期为 Delay(n-2), Delay(n-1),丢失帧后一帧的基音周期为 Delay(n+1),则构建丢失帧基音周期

的方法如下：

[0138] 首先, 判决定上升还是上降: 如果丢失帧前一帧的基音周期大于丢失帧后一帧的基音周期 (即 $\text{Delay}(n-1) - \text{Delay}(n+1) > 0$), 则判决为下降情况; 如果小于 0, 则为上升情景; 如果等于 0, 则为持平情景。

[0139] 其次, 判断变化趋势是快速还是平稳: 如果丢失帧前两帧的基音周期差值的绝对值 Delay_abs ($\text{Delay_abs} = |\text{Delay}(n-2) - \text{Delay}(n-1)|$) 大于等于设定的门限值 Delay_set , 则判断为快速变化, 否则, 判断为平稳变化; 门限值的设定并不限定, 可以根据实际情况进行调整。

[0140] 最后, 根据分类结果进行恢复, 具体的:

$$[0141] \quad \text{Delay}(n) = \begin{cases} \text{Delay}(n-1) \pm \frac{|\text{Delay}(n+1) - \text{Delay}(n-1)|}{2} & \text{(快速上升/下降)} \\ \text{Delay}(n-1) \pm \frac{|\text{Delay}(n+1) - \text{Delay}(n-1)|}{4} & \text{(快速上升/下降) (公式 2)} \\ \text{Delay}(n-1) & \text{(持平)} \end{cases}$$

[0142] 实际应用过程中, 由于解码出的基音周期分为整数基音周期 $T0$ 和分数基音周期 $T0_frac$ 两部分 (分数基音周期的变化单位为 $1/3$ 或 $1/6$ 个整数基音周期单位), 为了取得更好的构建效果, 在快速变化时, 还可以以整数基音为单位套用公式 2, 用 $T0$ 套用公式 2 后, 得到的公式如下:

$$[0143] \quad \text{Delay}(n) = \begin{cases} T0(n-1) \pm \frac{|T0(n+1) - T0(n-1)|}{2} & \text{(快速上升/下降)} \\ \text{Delay}(n-1) \pm \frac{|\text{Delay}(n+1) - \text{Delay}(n-1)|}{4} & \text{(快速上升/下降) (公式 3)} \\ \text{Delay}(n-1) & \text{(持平)} \end{cases}$$

[0144] 本发明实施例中, 构建丢失帧恢复自适应码书增益以及固定码书增益的方法与上述需要构建出丢失帧的基音周期的方法类型, 也是先判断上升或下降, 再判断平稳或快速, 最后参考前后帧参数进行构建, 在此不再赘述。

[0145] 由于丢失帧的类型不同, 因此, 需要构建的与其类型相符的特征参数也不同。参见图 6, 为本发明实施例 AMR-NB 基于不同的帧类型对丢失帧特征参数进行恢复的方法, 包括:

[0146] S351、判断帧的类型; 当为 RX_SPEECH_GOOD 时, 执行步骤 S352; 当为 RX_SID_FIRST 时, 执行步骤 S352'; 当为 RX_NO_DATA 时, 不进行丢帧恢复操作, 直接透传至后级, 由后级 (相对于丢失帧恢复这一级) 处理单元来完成相应的操作;

[0147] S352、构建 LSF 参数;

[0148] S353、构建基音延迟参数;

[0149] S354、构建自适应码书增益以及固定码书增益参数;

[0150] S352'、构建 LSF 参数, 与 S352 类似, 也构建 LSF 参数, 只是此处针对 RX_SID_FIRST 帧;

[0151] 从以上步骤可看到, 当帧类型为 RX_SPEECH_GOOD 时, 需要构建 LSF、基音延迟、自

适应码书增益以及固定码书增益等与该帧类型相符的特征参数；而当帧类型为 RX_SID_FIRST 时，只需构建 LSF 特征参数。其中，上述各参数的具体恢复方法在本实施例中相应部分已经进行描述，在此不再赘述。

[0152] 在构建与丢失帧类型相符的特征参数后，继续执行如下步骤：

[0153] S306、对丢失帧特征参数进行编码得到恢复后的新帧；

[0154] 即对构建的丢失帧的特征参数进行编码，得到恢复后的新帧；编码由系统的编码器来完成。

[0155] S307、执行正常帧处理流程。

[0156] 恢复后的新帧可认为是正常帧，后续执行正常帧的处理流程；当步骤 S302 判断没有帧丢失时，执行步骤 S308 后也执行该步骤。

[0157] 本发明实施例中，通过利用相邻帧的类型与特征参数来恢复丢失帧，由于解参数过程相比于完全解码到波形域的过程所消耗的资源要少得多，因此，通过本发明实施例恢复方法可以大大降低系统消耗的资源，在网络条件较差，存在很多丢失帧的情况下，这种效果更加明显。

[0158] 此外，本发明实施例还通过设定两种模式（延迟模式以及非延迟模式），在系统运行时，可以根据实际情况自动选择或者通过人工手动配置进入哪种模式，满足了不同场景下对传输质量及传输时延的要求。

[0159] 实施例四

[0160] 本发明实施例基于上述实施例对丢失帧进行恢复的基础上，通过结合其他参数域的语音增强技术以达到更好的语音增强效果。

[0161] 参见图 7，本发明实施例在基于丢失帧恢复的基础上，还通过其他参数域语音增强方法来实现语音增强，即对经过丢帧恢复后的语音帧再进行解参数，得到特征参数，根据得到的特征参数进行语音增强。

[0162] 如图 7 所示，这里本发明实施例应用在网络中间网元设备中的示意图，在该设备中，接收到的码流可分为上行码流以及下行码流。其中，上行处理过程中，在丢失帧恢复后，可以先对信号进行分类，然后根据分类结果执行后续 AEC、ANR、ACLIP 以及 ALC 各语音增强功能模块的处理流程。其中，上述各功能模块的含义如下：

[0163] AEC(Acoustic Echo Control, 自动回声控制)；

[0164] ANR(Automatic Noise Reduction, 自动噪声抑制)；

[0165] ACLIP(Anti-clip, 抗削波)；

[0166] ALC(Automatic Level Control, 自动电平控制)；

[0167] 需要说明的是，本发明实施例各语音增强步骤之间用虚线箭头连接仅表示后续可能执行这些步骤中的一个或多个（按箭头顺序），并不代表后续某个功能模块的输出是另一个的输入，如上行处理过程中，AEC 的输出并不代表 ANR 的输入。

[0168] 下行处理过程与上行处理过程类似，在进行丢失帧恢复后，先对信号进行分类，然后根据分类结果执行后续 ANR、ACLIP、ALC 以及 ANC 中的一个或多个功能模块的处理流程。其中，ANR、ALP、ALC 的定义在文已经介绍，在此不再赘述；这里的 ANC 表示 Automatic Noise Compensation，即自动噪声补偿。

[0169] 本发明实施例中，上下行的处理并不完全相同，例如，考虑到回声主要来自于终端

侧,因此,上行过程当中增加了 AEC 模块,而下行中并没有增加 AEC 模块。当然,实际使用中也可以使用相同的处理方法,在此并不限定。另外,上下行也可以使用其中一种或多种语音增强方法,如上行只使用 AEC,下行只使用 ACL 等。

[0170] 当上行或下行需要使用多种增强处理方式时,可以根据每个功能模块处理方式的特点对处理顺序进行排序。例如,AEC 用于回声控制,一般放在最前面,这样可以先去掉回声,否则,信号在处理过程当中始终会带着回声,在进行信号增强时会把回声也增强了,给后续回声控制增加难度;而 ALC 一般放在最后,这是因为 ALC 用于将语音的幅值调到目标范围内,如果放在前面,则后续其他模块处理后,又会引起语音幅值的变化,还得进行 ALC 调整,浪费了前面一次 ALC 处理过程。

[0171] 此外,还需要说明的是,上下行的数据可以是独立的,也可以是相关的(即其中有些数据需要相互交互),当上下行的数据是独立时,完全可以只使用上行模块或者下行模块;或者,如果实际应用只需要处理上行或下行数据,则本发明实施例可以只设置对应的上行或下行处理模块。

[0172] 为了更好地对信号分类进行说明,下面结合图 8 来进行详细阐述。参见图 8,为本发明实施例进行信号分类处理的一个示意图,本发明实施例中,输入的码流首先通过一个信号分类器,区分是静音、噪声、语音、音乐、回声以及信号音中的哪种,实际应用中,用户也可以添加其他自定义的分类,或者删除一些分类。信号分类器区分是哪种信号可以通过分析各种特征参数来得到,例如,如果分析到某些帧没有能量,则可以判决为静音;如果某些帧的参数符合话音帧或音乐帧的特征,则可以判决为话音帧或音乐帧。

[0173] 当由信号分类器确定是哪种类型后,送到相应的功能模块进行处理,例如,当确定是静音或噪声后,通过噪声电平跟踪后送到 ANR 功能模块进行处理;当确定是语音或音乐时,通过语音电平跟踪送至 ALC 以及 ANC 功能模块进行处理;需要说明的是,这里为了方便将噪声电平跟踪以及语音电平跟踪以单独的模块形式呈现,实际应用中,噪声电平跟踪以及语音电平跟踪都由分别由 ANC, ALC 或 ANC 内部实现,例如 ALC 和 ANC 每个单元内部都有一个语音电平跟踪功能模块来实现对语音电平的跟踪;当确定是回声时,直接送到 AEC 进行处理;当收到的是信号音时,不进行处理,直接输出。各功能模块具体处理步骤可以参考上述描述,在此不再赘述。

[0174] 本发明实施例中,与丢失帧恢复处理方法类似,每个功能模块的处理也在参数域进行,通过缓存的解参数后得到的相关特征参数并通过一定的算法进行恢复,算法实现过程中,也可以利用一些相邻帧的相关性并引入一定的经验值或影响因子,使得恢复结果更加准确。

[0175] 例如,以 ACLP(抗削波)处理为例,参见图 9,包括如下步骤:

[0176] S401、削波状态识别;

[0177] 具体的,可以基于缓存的帧得到出现削波帧的比例,当比例达到一定阈值,认为需要进行削波处理;其中,削波帧的判断方法通过解参数后得到的特征参数来判断,例如,可以通过固定码书增益或自适应码书增益等参数来判断能量是否溢出(即是否超过一定阈值),如果是,则判断为削波帧。

[0178] S402、判断当前帧是否为削波;

[0179] 在进行削波状态识别后,如果状态是需要进行削波处理,则后续针对每一帧判断

是否为削波判断后输出三种结果,分别为否、弱、以及强;

[0180] 如果为否,执行步骤 S403;如果为弱,执行步骤 S404;如果为强,执行步骤 S405;

[0181] S403、更新缓存的正常帧参数;

[0182] 如果为否,则进行透传处理,不需要恢复,更新完缓存的正常帧参数即可;

[0183] S404、削波恢复因子弱更新;后续执行步骤 S406;

[0184] 这里的削波恢复因子是指跟恢复有关的影响因子,类似于实施例三中在进行丢失帧 LSF 参数构建时公式中的影响因子 $fac()$,这里的削波恢复因子也是一个经验值,用于决定削波恢复的程度,例如,如果削波增益只超阈值一点点,则恢复时也只需要将削波增益减少一点点,此时,削波恢复因子就可以设得平稳一些(即变化不需要太大);反之,如果削波情况比较严重,恢复时需要将削波增益大大减少,则此时需要将削波恢复因子设得剧烈一些(即变化需要大一点)。

[0185] 当判断为“弱”时,表示虽然有削波发生,但不是非常严重,因此,弱更新时,削波恢复因子可以选择一个比较平稳的值。

[0186] S405、削波特征参数强更新;后续执行步骤 S406;

[0187] 与判断为“强”时,表示削波情况比较严重,因此,需要通过强更新来将削波恢复因子设置成一个比较剧烈的值。

[0188] S405、进行削波恢复;

[0189] 削波恢复因子更新后,便可以根据削波恢复因子以及解参数后得到的一些特征参数(如自适应码书增益、固定码书增益等)来进行削波恢复。通过上述步骤,即可以得到进行削波恢复,在进行其他处理时(如 AEC、ALC),也可以利用参数域中的特征参数并结合一些影响因子通过一定的算法来实现各种对应的功能。

[0190] 本发明实施例通过对丢失帧恢复的基础上,通过分类器判断信号属于哪种类型,并进行相应的处理,可以针对不同类型进行相应的处理,从而进一步提升了语音质量。

[0191] 通过本发明实施例,可以实现对语音进一步的增强,同时,由于也在参数域进行语音增强,因此,也能降低系统消耗的资源;结合前面实施例中的丢帧恢复技术,能取得更好的效果。

[0192] 同时,本发明实施例当有多个语音增强模块(如 AEC、ANC 等)时,可以根据实际情况对各增强单元在顺序上进行调整,并且通过信号分类器对信号进行划分,以取得更好的效果。

[0193] 实施例五、

[0194] 本发明实施例提供了一种丢帧恢复设备 51,用于对丢失帧进行恢复,参见图 10,包括;

[0195] 帧信息确定单元 511,用于根据与丢失帧相邻的相邻帧类型确定丢失帧的类型;

[0196] 解参数单元 512,用于获取对相邻帧进行解参数后得到的特征参数,该特征参数用于确定帧中编码数据的特征信息;

[0197] 特征参数构建单元 513,用于根据相邻帧的特征参数以及多个帧中编码数据特征信息之间的相关性,构建与丢失帧类型相符的特征参数;

[0198] 恢复单元 514,用于根据构建的丢失帧的特征参数恢复丢失帧。

[0199] 参见图 11,具体的,上述帧信息确定单元 511 还可以包括丢失帧确定单元 5111、帧

类型确定单元 5112 以及丢失帧类型确定单元 5113 ;其中,

[0200] 丢失帧确定单元 5111 用于,通过检测收到的帧中的帧序号,或者时间戳确定是否符合预设规则的方法来确定丢失帧及与相邻帧;

[0201] 帧类型确定单元 5112 用于,通过帧中相关标志位确定帧的类型;

[0202] 丢失帧类型确定单元 5113 用于,根据相邻帧类型确定丢失帧类型,其中,相邻帧的类型可由帧类型确定单元 5112 来完成,而丢失帧的确定可由丢失帧确定单元 5111 来完成;

[0203] 参见图 12,上述语音增强设备还可以包括:

[0204] 模式选择单元 515,用于当丢失帧确定单元确定有丢失帧且需要恢复时,判断是否进入延时模式,如果是,则特征参数构建单元 513 使用前后帧的特征参数进行恢复,此时,相邻帧包括前后帧;否则,进入非延时模式,特征参数构建单元 513 使用前帧的特征参数进行恢复,此时,相邻帧指前帧。

[0205] 上述各单元具体执行的步骤可以参见上述实施例一、二、三中的相关描述,本领域技术人员可以从以上实施例中描述来实现本发明实施例各单元具体执行步骤。

[0206] 参见图 13,本发明实施例还可以提供了一种语音增强设备,包括丢帧恢复设备 51,还包括以下几个语音增强单元中的一个或多个(也可以全部):

[0207] 自动回声控制(AEC)单元 52,用于自动回声控制;

[0208] 用于自动噪声抑制(ANR)单元 53,用于自动噪声抑制;

[0209] 抗削波(ACLIP)单元 54,用于抗削波;

[0210] 自动电平控制(ALC)单元 55,用于自动电平控制;

[0211] 自动噪声补偿(ANC)单元 56,用于自动噪声补偿。

[0212] 当有多个单元时,还可以包括:

[0213] 分类单元 57,用于对输入的信号进行分类,确定信号的类型,并根据确定的类型送至上述一个或多个对应的功能单元进行处理。如果只使用其中一个单元,则可以不使用分类单元,直接根据信号的类型进行相应处理。

[0214] 分类单元 521 及上述 AEC、ANR、ACLIP、ALC、ANC 等单元都是基于参数域进行信号处理,即对码流进行解参数后得到特征参数,根据特征参数对信号进行自动回声控制、自动噪声抑制、抗削波、自动电平控制或者自动噪声补偿中的一种或多种语音增强处理。其具体处理方法可以参见实施例四中的相关说明,本领域技术人员可以根据实施例四中描述来实现各单元的功能,在此不再赘述。

[0215] 在具体硬件实现上,本发明实施例可以各单元可以通过通用处理器(如 DSP、FPGA 等)来实现,也可以采用 ASIC 专用电路来实现,在此并不限定。同时,如果本发明实施例中设备可以是独立的设备,也可以与其他设备进行集成,例如,如果原来有设备中的相应解参数单元(如通过自带解码器实现解参数功能)也可以完成解参数的功能,则本发明实施例解参数单元也可以采用原有设备中的解码单元来完成解参数;如果是独立的设备,则可以通过设计相应的解参数模块来完成需要的功能。

[0216] 参见图 14,为本发明实施例中一个具体实现的硬件示意图,包括 DSP、存储器、接口单元、附属电路等部分;其中,DSP 为主要处理芯片,用于运行处理代码,实现上述各单元的功能;存储器用于为程序运行提供所需的存储空间,如 DDR、flash 等;接口单元用于该设

备与其他设备进行通信,如通过串口、以太网或其他接口进行通信;附属电路可以是系统运行所需一些电路,如电源电路,晶振电路、监控电路等,在此不再赘述。

[0217] 在图 14 所示的实施例中,DSP 负责所有的程序代码执行,包括对码流进行解参数;在另一个实施例中,如果系统内的其他设备已经带有解参数功能的话,那么 DSP 可以直接使用该设备解参数后得到的特征参数,而不需要自己再执行解参数。参见图 15,为本发明另一实施例 DSP 直接使用另一设备(如编解码器,也可以用 DSP 实现)解参数后输出的特征参数示意图。

[0218] 本发明实施例通过在参数域进行丢失帧恢复,可以降低系统消耗的资源,同时,通过模式选择单元选择不同的模式进行恢复,可以满足系统对延时或质量的要求;并且,本发明实施例还可以加入其他语音增强单元,以达到更好的语音增强效果。

[0219] 实施例六

[0220] 本发明实施例提供了一种语音增强系统,参见图 16,包括:

[0221] 第一基站 61,第一终端 62;第二基站 63,第二终端 64;丢帧恢复设备(或语音增强设备)65;

[0222] 其中,第一基站 61 与第一终端 62 之间通过第一链路 66 进行连接;第二基站 63 与第二终端 64 之间通过第二链路 67 进行连接;第一基站 61 与第二基站 64 之间通过第三链路 68 进行连接;

[0223] 其中,丢帧恢复设备(或语音增强设备)65 为上述实施例中提供的丢帧恢复设备或语音增强设备(以下统一称为语音增强设备,并不对两者进行严格区分),用于对第一链路、第二链路或者第三链路中的一个或多个环节出现的丢帧进行恢复;或者还可以对码流进行解参数后得到的特征参数,根据特征参数对信号进行自动回声控制、自动噪声抑制、抗削波、自动电平控制或者自动噪声补偿中的一种或多种语音增强处理。

[0224] 具体的,语音增强设备根据应用场景不同位于系统中的位置也不同,下面通过几种常用的场景进行介绍。

[0225] 参见图 17,为全 IP 组网下 TrFO(Transcoder Free Operation,免编解码级联操作)场景应用示意图,包括 IP 网(图中 Internet)以及位于 IP 网两侧的基站和终端。TrFO 的特点在于通信双方(如图中两个终端)采用相同的编码形式,因此,在传输过程当中不需要进行编码转换,都是以码流的形式进行传输(不包括通过 Internet 传输时打包),这些码流都可以作为本发明实施例中的输入。

[0226] 如图 17 所示,在网络传输过程中,会有几个丢包(丢帧)环节,如丢包环节 1 表示在 IP 网络出现丢包;丢包环节 2 表示终端跟基站上行或下行也会出现丢包的情况。

[0227] 如图 17 所示,为了解决 IP 网络中的丢包环节 1 导致的丢包情况,将语音增强设备放在位于基站至 IP 网中的相关网元中,例如,放在 BSC(Base Station Controller,基站控制器)或核心网中的相关网元;

[0228] 或者,也可以将语音增强设备放在基站或终端中,以解决丢包环节 2 导致的丢包情况;

[0229] 或者也可以在几个位置都增加语音增强设备。

[0230] 参见图 18,为另一应用语音增强设备的系统示意图,图 18 所示的应用场景为 TF0(Tandem Free Operation,免级联操作),在 TF0 应用场景中,两侧终端对应的编码类型

也相同,在传输过程中不需要进行编码转换,都以码流的形式进行传输,这些码流也可以作为语音增强设备的输入。

[0231] 其中,丢包环节可能发生在终端与基站上下行传输过程当中,语音增强设备的位置并不固定,例如可以位于终端或基站上,解决上下行丢包的问题;或者也可以如图 18 所示,位于两个基站之间,解决其中一侧产生的丢包问题;或者也可以在几个位置都增加语音增强设备。

[0232] 参见图 19,为本发明另一实施例应用语音增强设备的系统示意图,图 19 所示的应用场景为 TDM(Time Division Multiplex,时分复用)传统组网。在 TDM 应用场景中,两侧终端的编解码类型可能不同(如一侧是 GSM,另一侧是 CDMA),此时,需要通过编解码器来对不同类型的码流进行转化,如图中 TC 所示。此时,语音增强设备位于 TC 的前端,经过处理后输出给 TC 的都是经语音增强设备处理(丢帧恢复及其他语音增强处理)过的好帧。

[0233] 当然,与前面几个应用场景类似,语音增强设备也可以位于基站或终端或者几个地方都进行部署。

[0234] 通过上述应用场景可知,本发明实施中的语音增强设备可以位于全 IP 组网下的 TrFO、以及 TFO、TDM 等应用场景,具有非常好的适用性;同时,通过本语音增强设备,可以基于输入的码流解决不同环节出现的丢包问题,提升了语音质量,并且与现有技术基于波形的恢复方法相比,大大降低了消耗的资源。

[0235] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory, ROM)或随机存储记忆体(Random Access Memory, RAM)等。

[0236] 上列较佳实施例,对本发明的目的、技术方案和优点进行了进一步详细说明,所应理解的是,以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

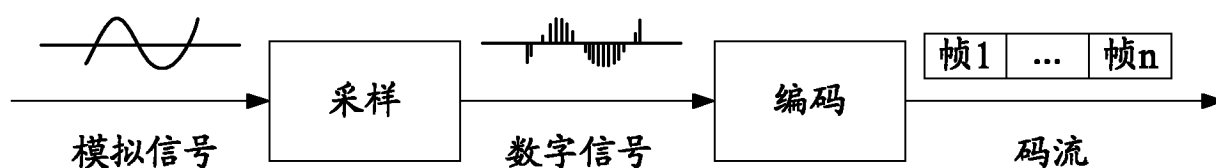


图 1

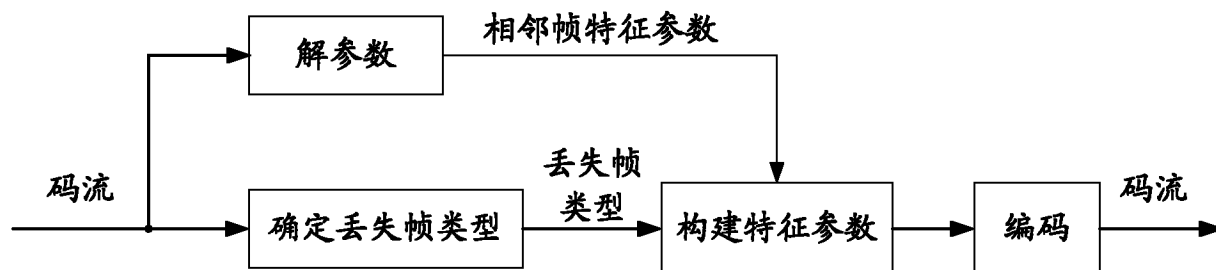


图 2

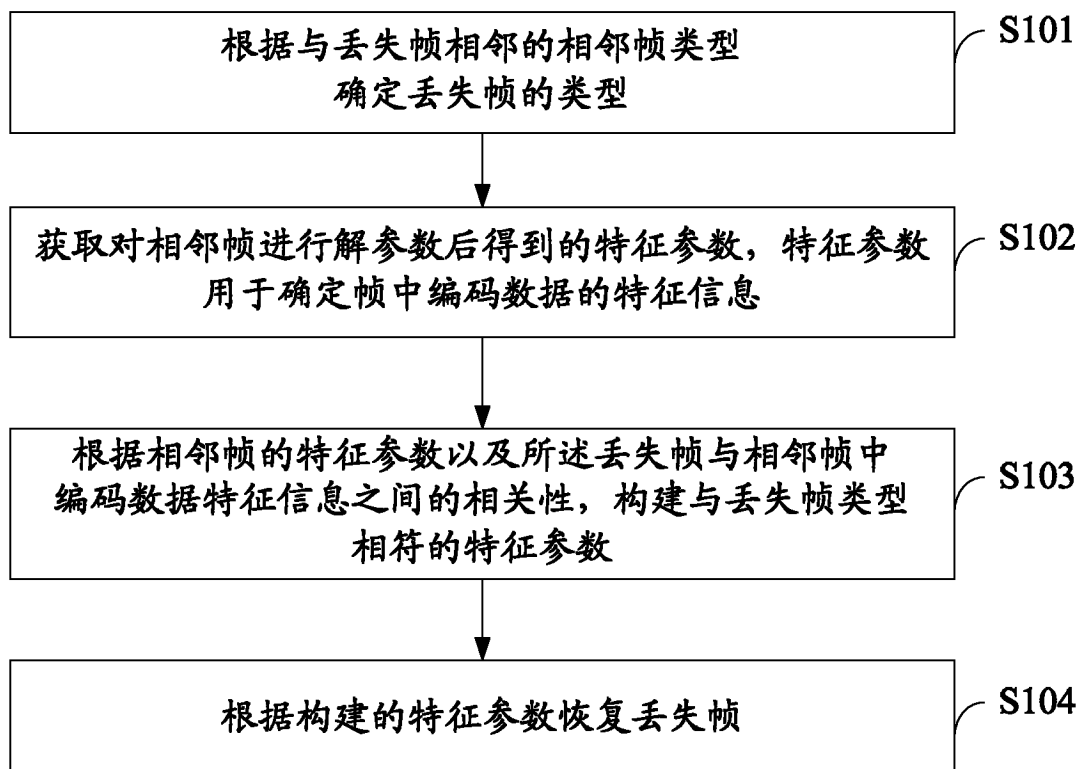


图 3

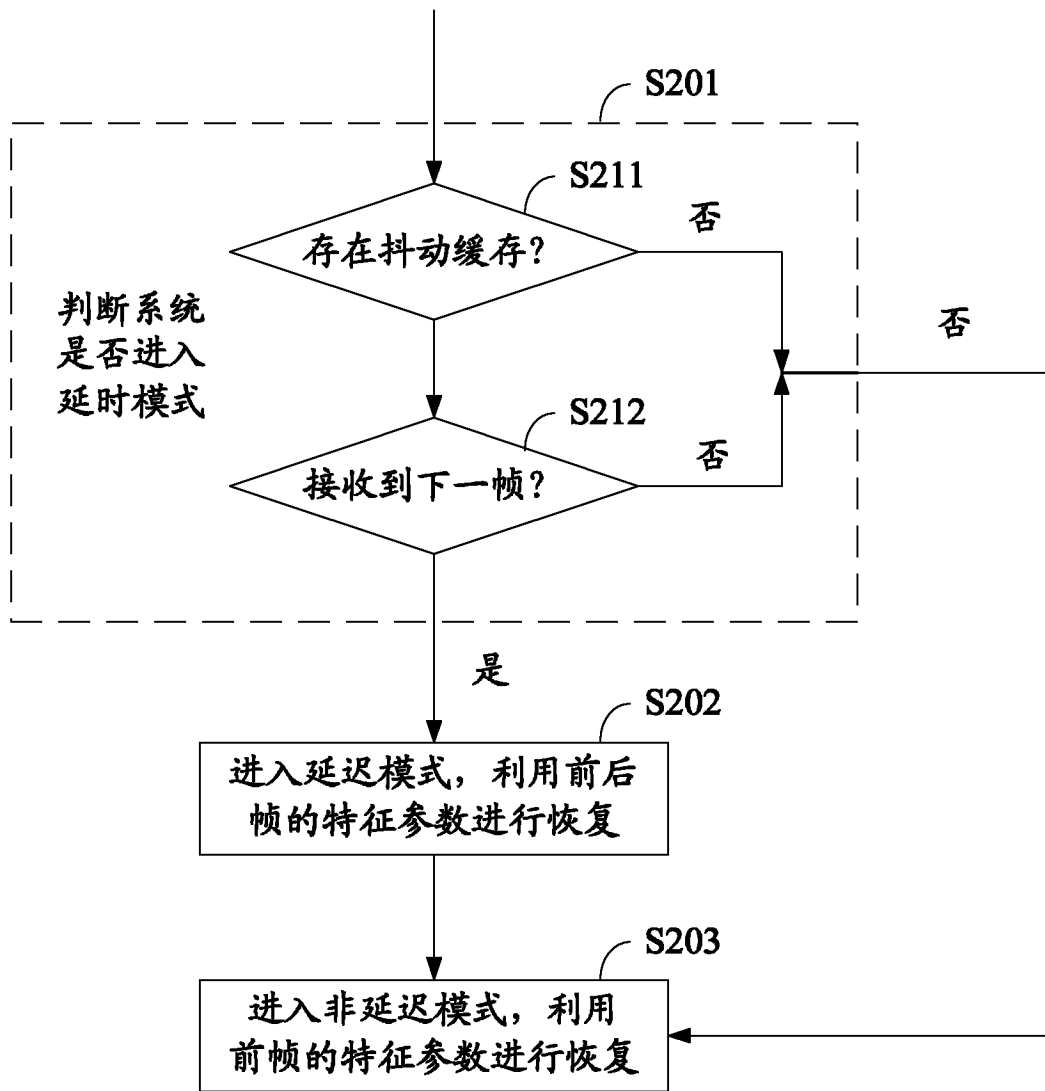


图 4

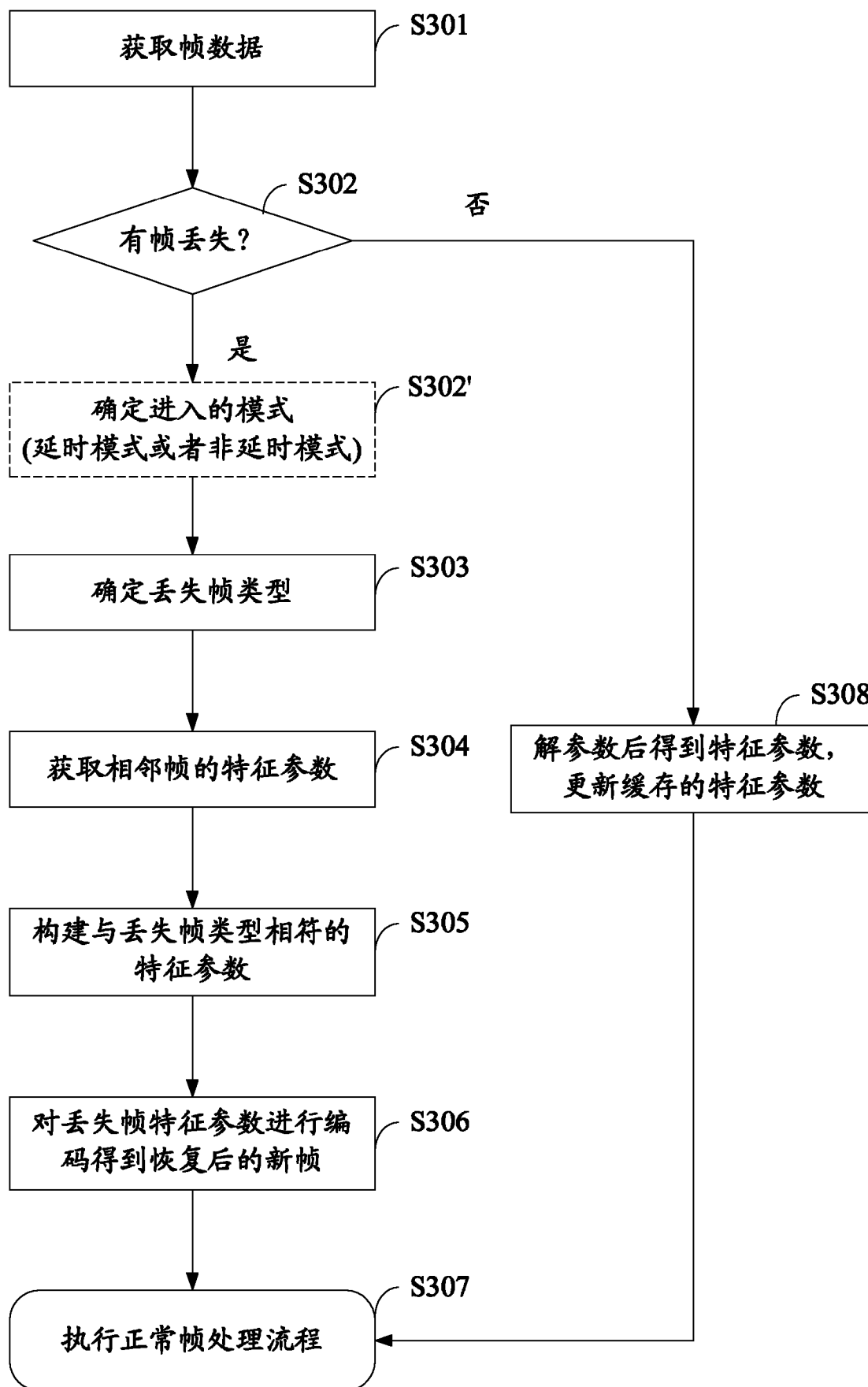


图 5

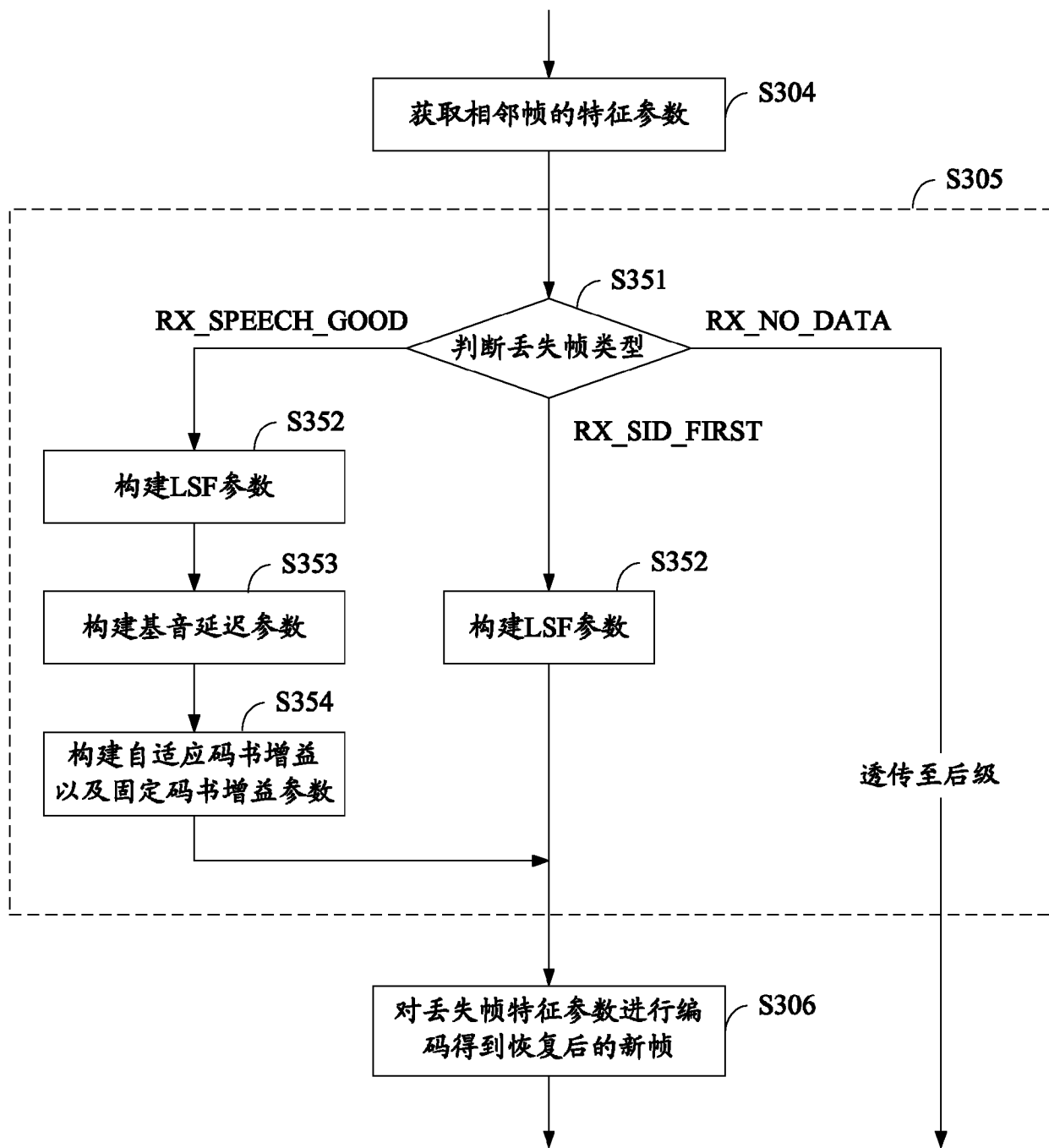


图 6

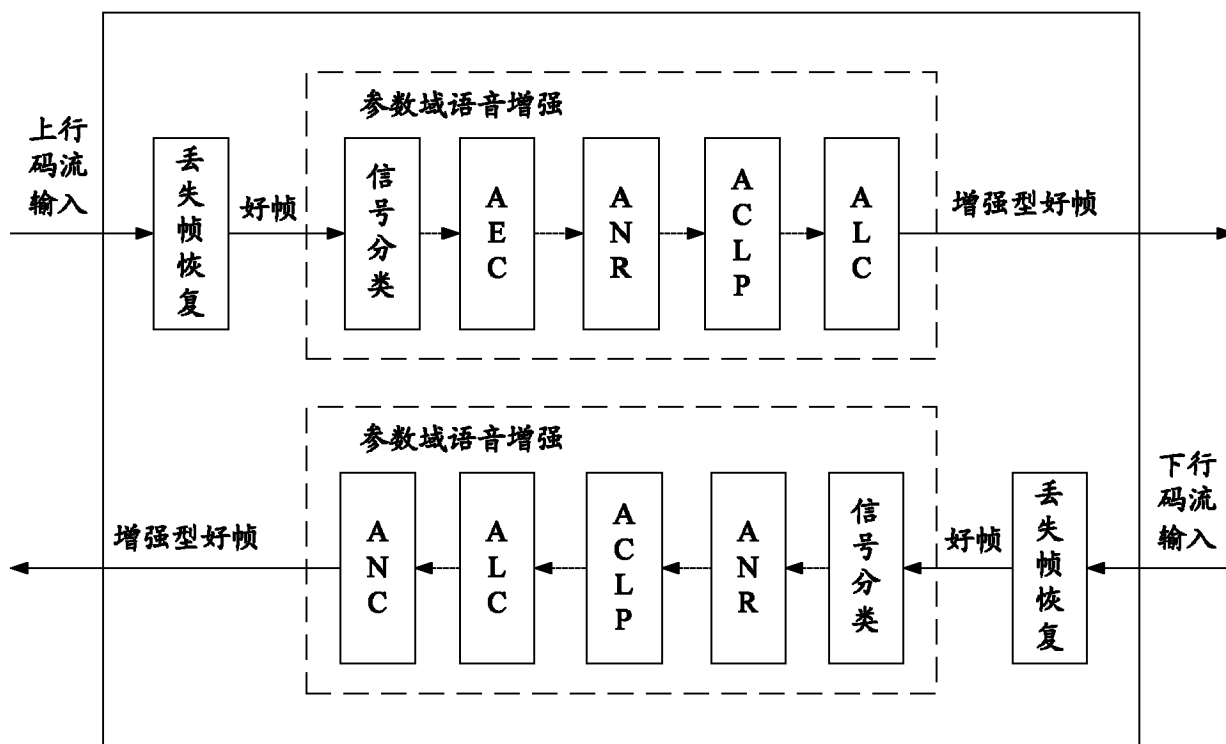


图 7

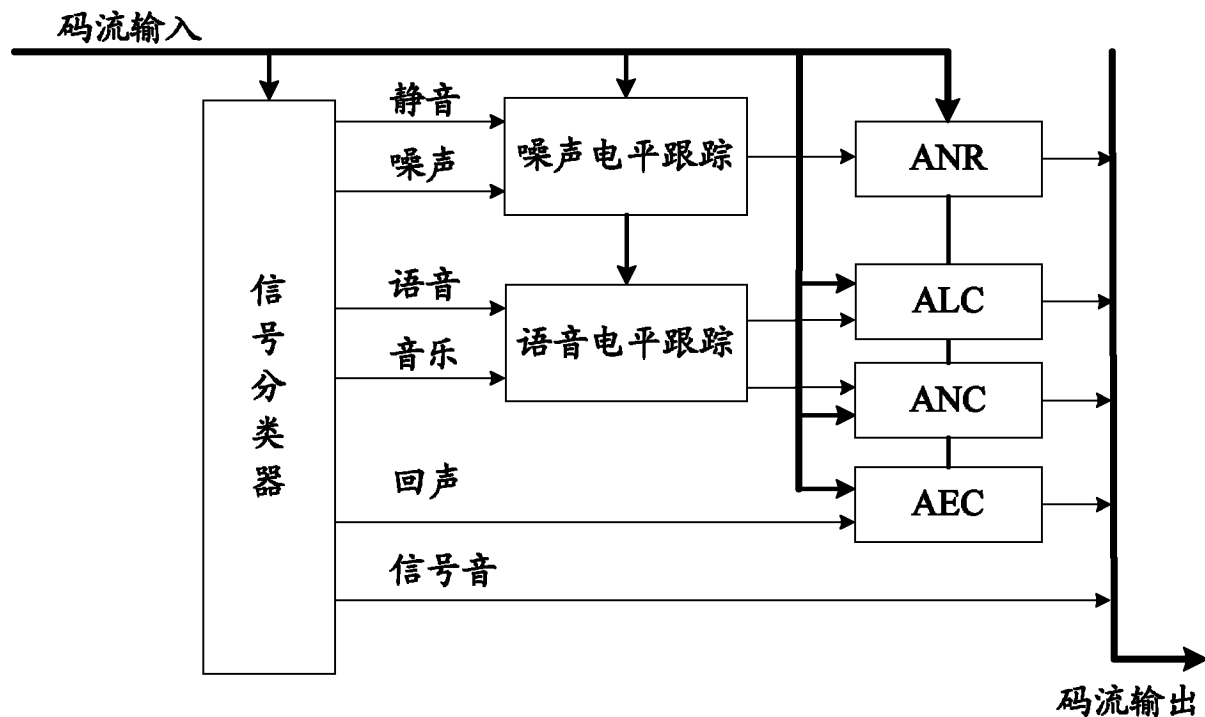


图 8

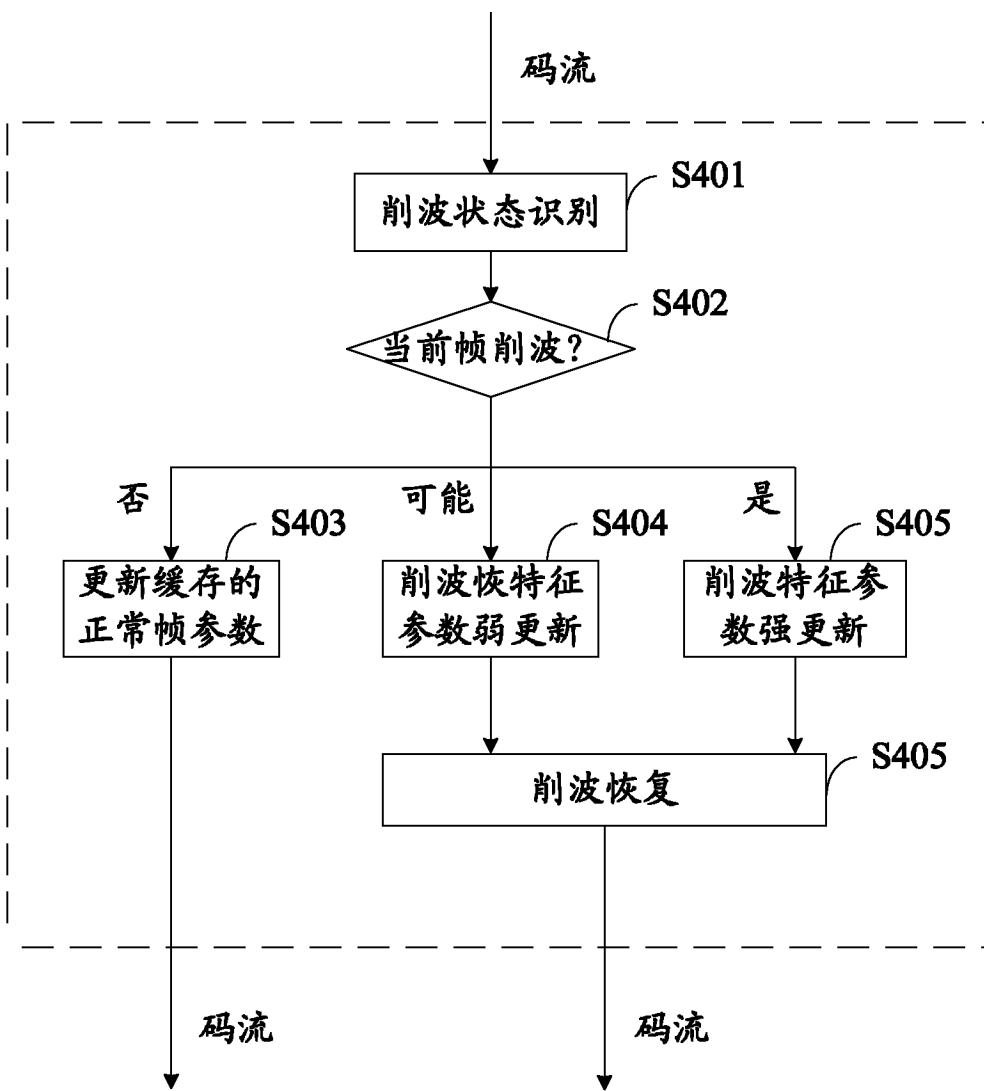


图 9

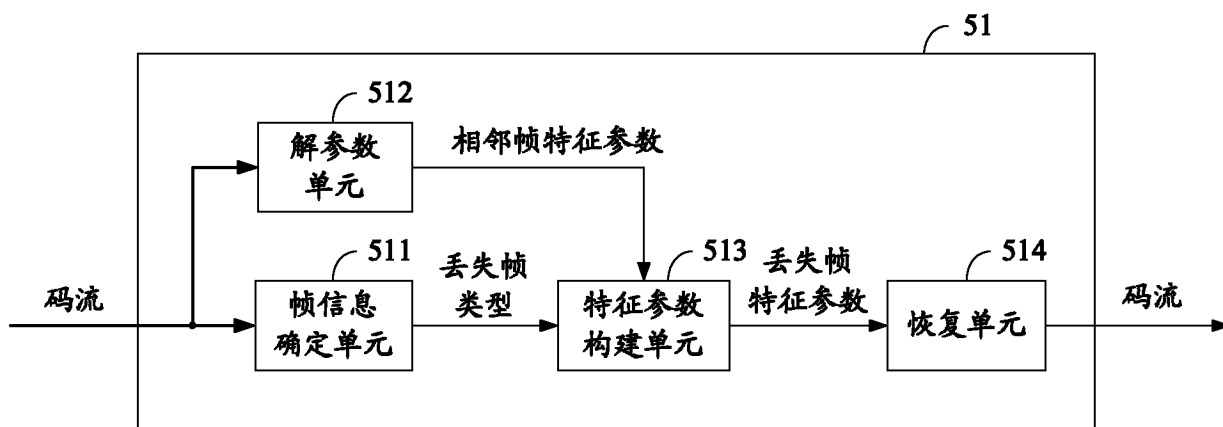


图 10

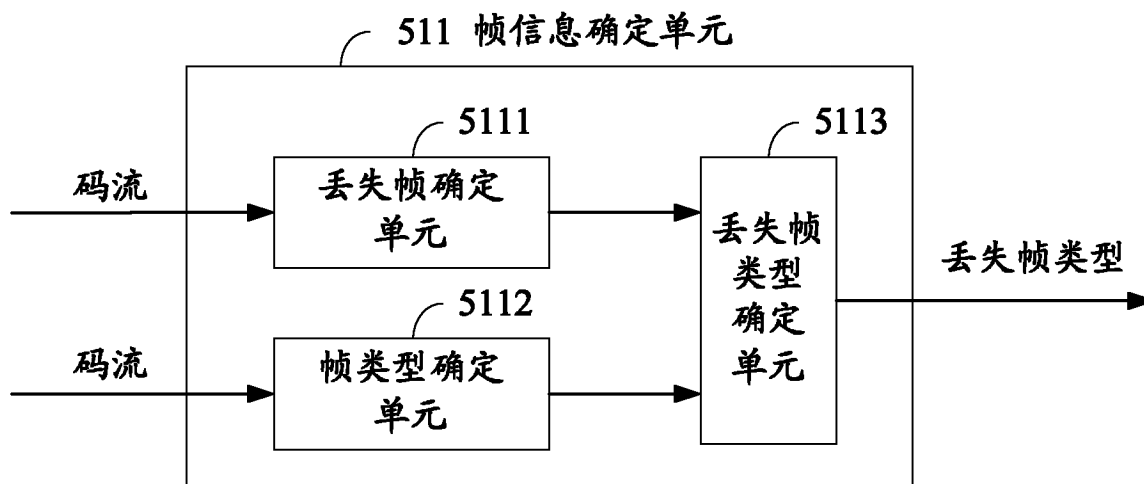


图 11

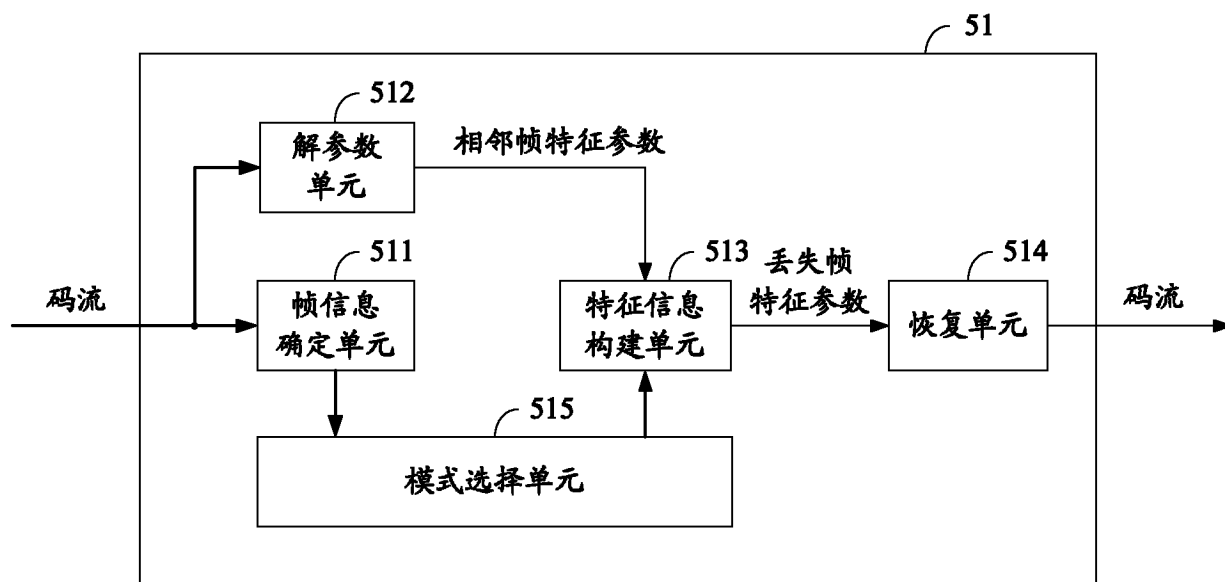


图 12

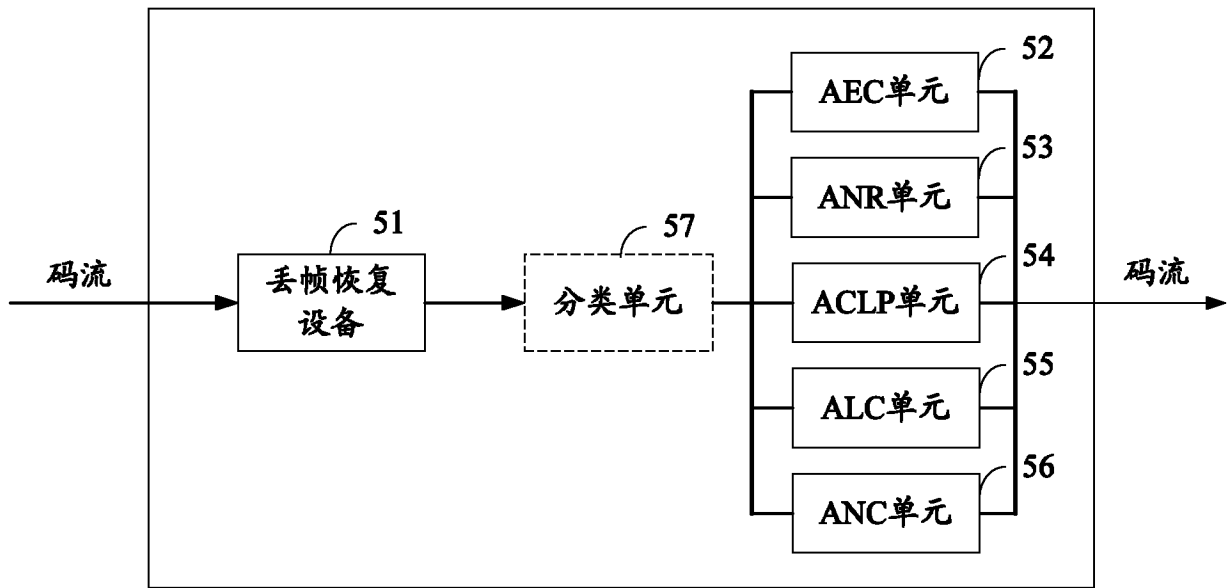


图 13

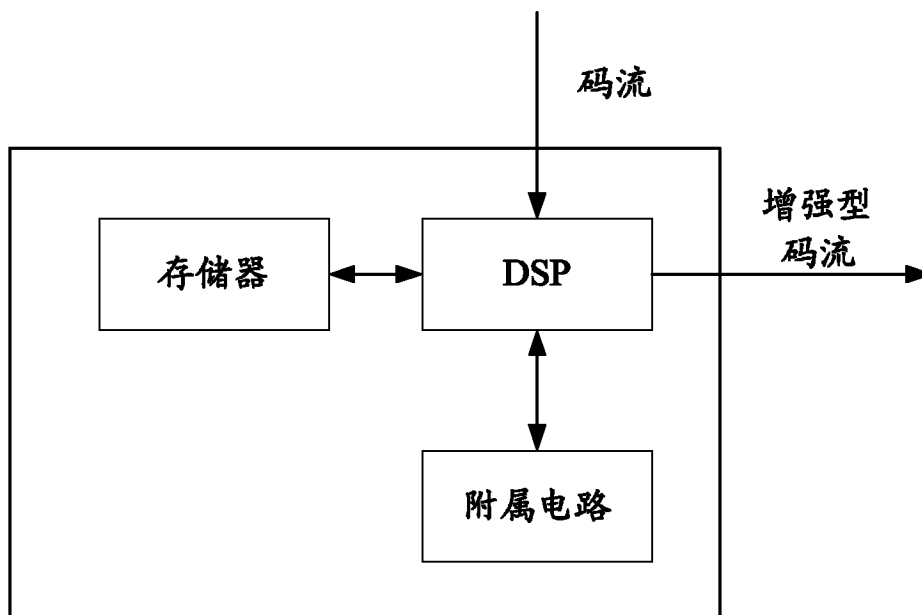


图 14

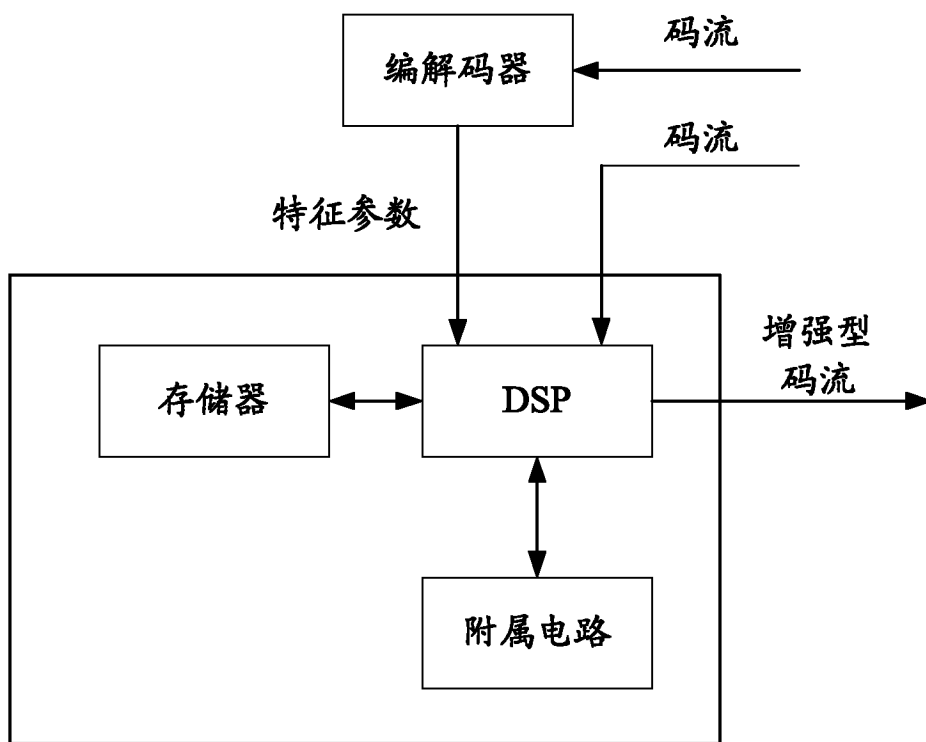


图 15

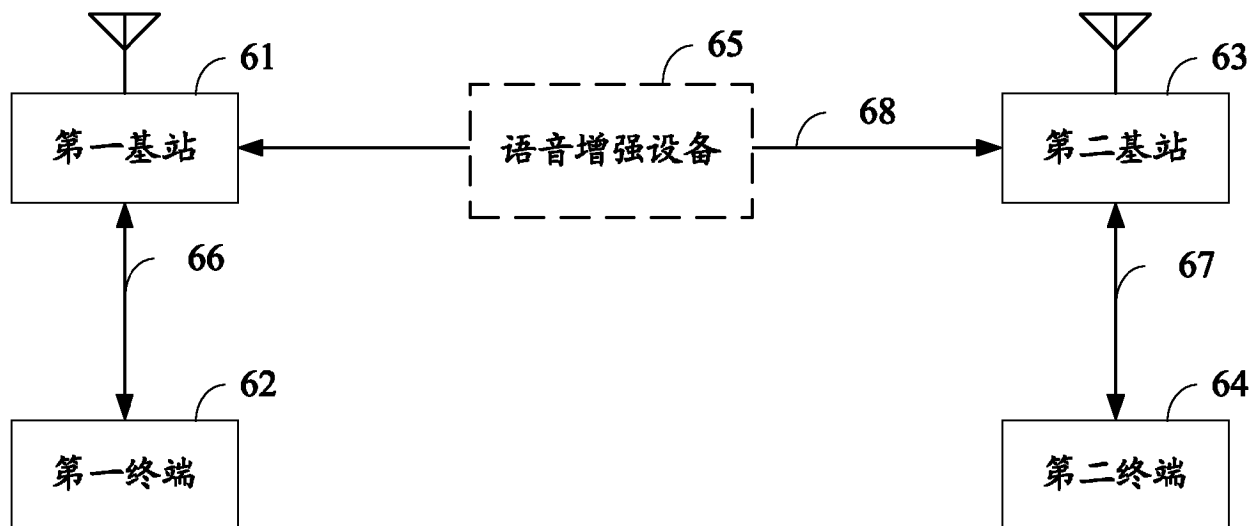


图 16

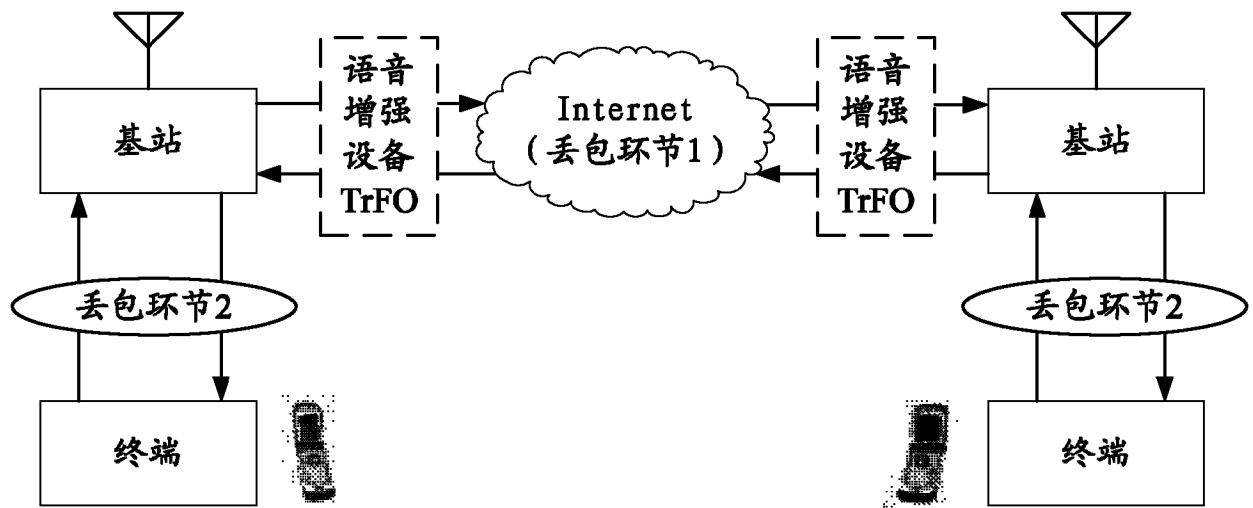


图 17

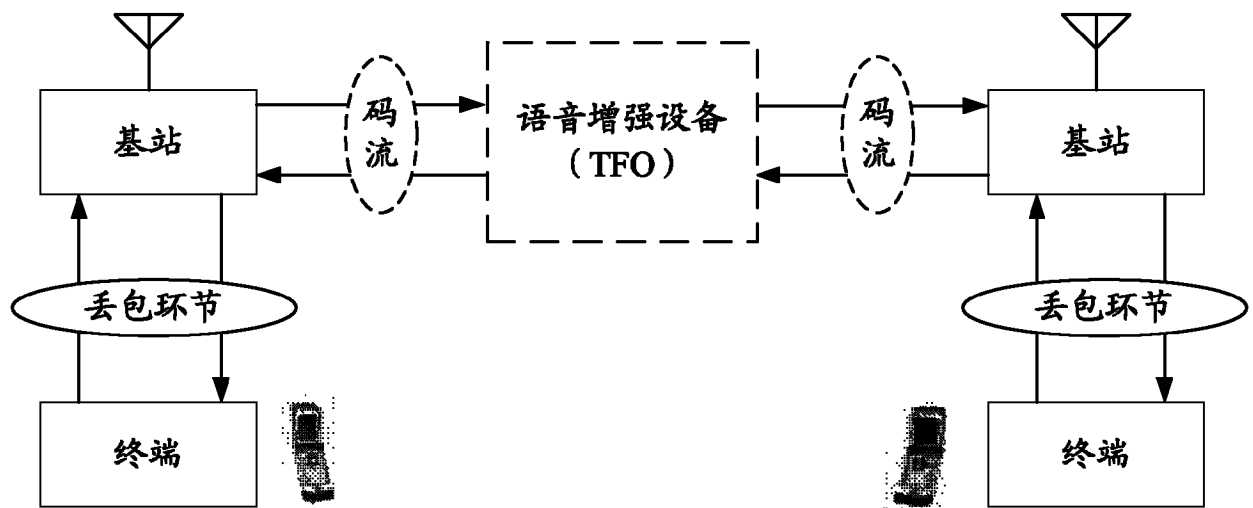


图 18

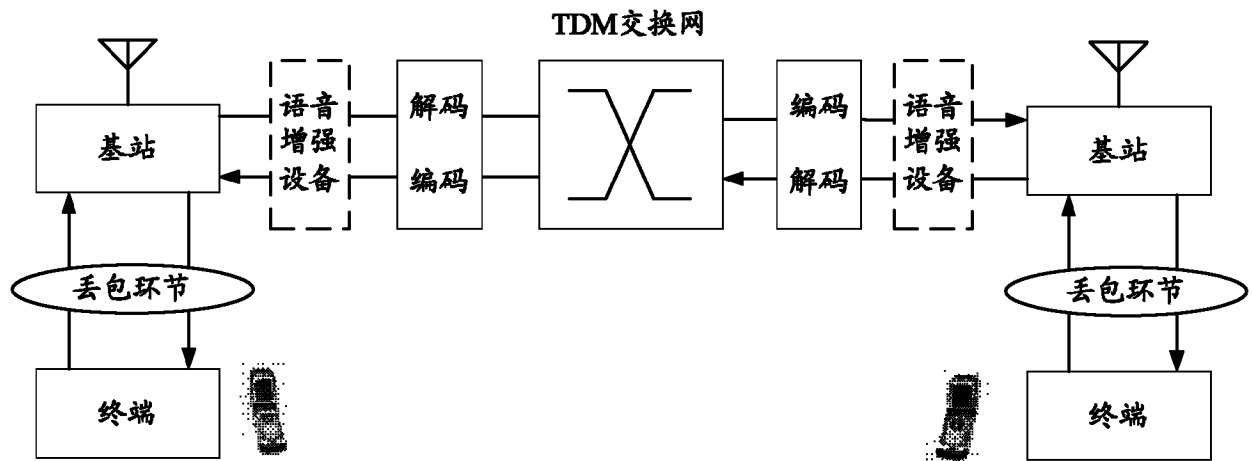


图 19