(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102263960 A
(43)申请公布日 2011.11.30

(21)申请号 201110232486.8

(22)申请日 2011.08.15

(71)申请人 复旦大学
地址 200433 上海市杨浦区邯郸路220号

(72)发明人 沈沙 钟慧波 刘家良 沈蔚炜
范益波 曾晓洋

(74)专利代理机构 上海正旦专利代理有限公司
31200
代理人 陆飞 盛志范

(51)Int.Cl.
H04N 7/26(2006.01)

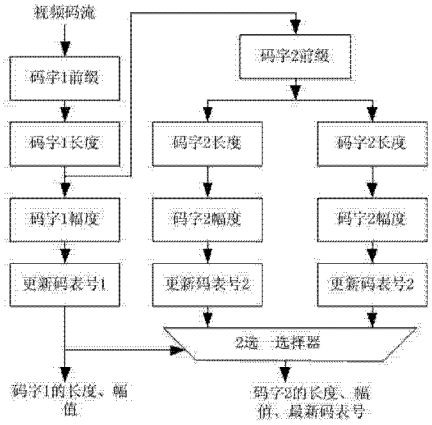
权利要求书 1 页 说明书 4 页 附图 1 页

(54)发明名称

一种快速解码CAVLC非零系数幅值的方法

(57)摘要

本发明属于数字视频信号编解码技术领域，具体为一种快速解码CAVLC非零系数幅值的方法。本方法将用于解码非零系数幅值的候选码表数目从7个降低到2个，在得到第一个码字的长度后，即可开始计算第二个码字。在对第二个非零系数幅值的码字进行解码时采用2路并行解码的方法，对可能采用的两个码表同时进行计算，得到两个可能的解码结果。当第一个码字的幅值和码表信息更新完毕，即可从两个结果中选择出正确的结果。这样可以缩短解码流程的关键路径，从而实现在一个时钟周期内同时解码两个非零系数的幅值。由于非零系数幅值的码字是CAVLC编码中出现频率最多的码字，这一方法大大提高了硬件解码器的数据处理能力。



CN 102263960 A

1. 一种快速解码 CAVLC 非零系数幅值的方法，其特征在于具体步骤如下：

（1）针对第一个码字，计算其幅值前缀的码字长度和解码后的值；

（2）计算第一个码字的长度；

（3）计算第一个码字的幅值；

（4）针对第第二个码字，计算其幅值前缀的码字长度和解码后的值；

（5）计算第二个码字的两种可能长度和幅值；

（6）更新第一个码字输出的码表号；

（7）针对第二个码字，选择正确的幅值和长度作为输出；

（8）更新第二个码字输出的码表号。

2. 根据权利要求 1 所述的快速解码 CAVLC 非零系数幅值的方法，其特征在于所述更新码表号方法如下：

按照 H.264 CAVLC 规定的 7 个可能的码表号：0、1、2、3、4、5、6，根据当前的码表号预测出可能用于下一个码字解码的新码表号：如果当前的码表号为 0，那么新码表号为 1；如果当前的码表号为 1，新的码表号有两个候选值：1、2；如果当前的码表号为 2，新的码表号有两个候选值：2、3；如果当前的码表号为 3，新的码表号有两个候选值：3、4；如果当前的码表号为 4，新的码表号有两个候选值：4、5；如果当前的码表号为 5，新的码表号有两个候选值：5、6；如果当前的码表号为 6，新的码表号为 6。

3. 根据权利要求 2 所述的快速解码 CAVLC 非零系数幅值的方法，其特征在于在计算第二个非零系数的幅值和码字长度时，采用 2 路并行的方法提前开始解码，即根据两个不同的候选码表号提前计算第二个码字的长度和幅值。

4. 根据权利要求 3 所述的快速解码 CAVLC 非零系数幅值的方法，其特征在于第一个码字的幅值前缀的值通过计算当前视频码流码流中前导 0 的个数来得到，幅值前缀的值等于前导 0 的个数；幅值前缀的码字长度等于幅值前缀的值再加 1。

5. 根据权利要求 4 所述的快速解码 CAVLC 非零系数幅值的方法，其特征在于第一个码字的长度等于幅值后缀的长度加上幅值前缀的长度；而解码幅值后缀的长度由幅值前缀和当前码表号决定：当幅值前缀小于 14 时，幅值后缀的长度等于码表号，当幅值前缀等于 14 时，幅值后缀的长度是 4 比特，当幅值前缀大于 14 时，幅值后缀的长度等于幅值前缀的值减 3。

6. 根据权利要求 5 所述的快速解码 CAVLC 非零系数幅值的方法，其特征在于所述

更新第一个码字输出的码表号的方法为：如果码表号等于 6，那么新的码表号为 6；如果码表号是 0，那么新的码表号是 1；否则，新码表号依照码字幅值的大小来决定是否等于当前码表号还是当前码表号加 1；同时，根据第一个码字输出的新码表号，从步骤（5）中得到的两个候选结果中选择正确的幅值和码字长度作为第二个非零系数码字的解码结果。

一种快速解码 CAVLC 非零系数幅值的方法

技术领域

[0001] 本发明属于数字视频信号编解码技术领域，具体涉及一种解码 CAVLC 非零系数幅值的方法。

背景技术

[0002] H.264/AVC(Advanced Video Coding) 由国际电信组织（ITU）和运动图像专家组（MPEG）联合制定而成的国际视频编码标准，目前已经在多媒体音视频领域得到了广泛的应用。H.264/AVC 中规定其熵编码可以采用两种方式：下文自适应可变长编码（CAVLC）和上下文自适应算术编码 (CABAC)。CABAC 具有较高的编码效率，但是编解码的复杂度也大大增加，而 CAVLC 在编码效率和复杂度上有着较好的均衡。与前一代视频标准如 MPEG-1, MPEG-2 相比，CAVLC 的编码效率有着显著提升。

[0003] 视频码流中的视频头信息和预测信息一般采用较低复杂度的定长码或指数哥伦布码，而残差系数的信息占据了视频码流的绝大部分，这部分信息的编码需要采用更高效的 CAVLC 编码方式。CAVLC 的码字总共有 5 类：

1. Coeff_token：本码字代表非零系数的数目和拖尾 1 的个数（TrailingOnes）。

[0004] 2. Sign_trail：本码字是拖尾 1 的符号。每一个符号编码为一比特，0 表示正 1，1 表示负 1。

[0005] 3. Levels：本码字是指 TrailingOnes 外的其它的非零系数的幅值。

[0006] 4. Total_zeros：它编码在以反 Zig-Zag 扫描顺序时第一个非零系数后的总的零系数的个数。

[0007] 5. Run_before：本码字是指每一个非零系数前的零的个数。

[0008] 对于硬件解码而言，计算非零系数的幅值是整个 CAVLC 解码过程中最耗费时间的过程。对于一个长度和高度都为 4 个像素的图形块，如果采用 CAVLC 编码方式，有些码字如 Coeff_token、sign_trial 和 Total_zero 在解码过程中只需要一次解码，而非零系数幅值的码字最多可以达 16 个。同时，由于 CAVLC 引入了上下文自适应的编码方式，当前的非零系数的幅值不仅取决于码字本身，还取决于前一个码字的解码结果，解码时只能按照顺序依次解码每一个非零系数幅值的码字，无法通过单纯增加硬件并行度来提升解码速度。

发明内容

[0009] 本发明的目的在于提出一种快速解码 CAVLC 非零系数幅值的方法，适用于 H.264 视频规范所规定的 CAVLC 解码过程，可以在一个时钟周期内完成两个非零系数幅值的解码。在解码得到第一个码字的长度后，第二个码字的解码过程就可以开始，不必等到第一个码字的所有信息都完成解码。解码过程的输入是待解码的视频码流和前一次解码中用的码表号，输出是两个非零系数的幅值、长度和使用的码表号。具体的解码过程如下：

（1）针对第一个码字，计算其幅值前缀的码字长度和解码后的值（简称幅值前缀的值）；

（2）计算第一个码字的长度；

（3）计算第一个码字的幅值；

（4）针对第第二个码字，计算其幅值前缀的码字长度和解码后的值；

（5）计算第二个码字的两种可能长度和幅值；

（6）更新第一个码字输出的码表号；

（7）针对第二个码字，选择正确的幅值和长度作为输出；

（8）更新第二个码字输出的码表号。

[0010] 本发明中，通过分析 H.264 视频标准，优化了码表号的更新过程，将候选的码表号由 7 个减少到 2 个。H.264 CAVLC 规定了 7 个可能的码表号，分别为 0、1、2、3、4、5、6。本方法根据当前的码表号预测出可能用于下一个码字解码的新码表号。如果当前的码表号为 0，那么新码表号为 1；如果当前的码表号为 1，新的码表号有两个候选值：1、2；如果当前的码表号为 2，新的码表号有两个候选值：2、3；如果当前的码表号为 3，新的码表号有两个候选值：3、4；如果当前的码表号为 4，新的码表号有两个候选值：4、5；如果当前的码表号为 5，新的码表号有两个候选值：5、6；如果当前的码表号为 6，新的码表号为 6。

[0011] 本发明中，消除了前后码字的相关性，在计算第二个非零系数的幅值和码字长度时，采用 2 路并行的方法提前开始解码，根据两个不同的候选码表号提前计算第二个码字的长度和幅值，而不是等到第一个码字完全解码结束后才开始，这样可以缩短硬件时序上的关键路径，加快解码流程。

[0012] 本发明的有益效果：

现有方法在一个时钟周期内只能解码一个非零系数的幅值，而本发明中描述的方法可以将速度加倍，在一个时钟周期内可以解码得到两个非零系数的幅值，从而提高了 CAVLC 整体的解码速度。适用于各种具有 H.264 视频解码功能的电子设备。

附图说明

[0013] 图 1：幅值后缀长度的计算过程。

[0014] 图 2：快速解码两个 CALVC 非零系数幅值的整体框图。

具体实施方式

[0015] 下面结合附图对本发明做进一步的描述。

[0016] 本发明所述的快速解码非零系数幅值的具体实施方式如下：

（1）计算第一个码字的幅值前缀（level_prefix）。幅值前缀的计算方法如表 1 所示。幅值前缀的值可以通过计算当前视频码流码流中前导 0 的个数来得到，幅值前缀的值等于前导 0 的个数。幅值前缀的码字长度等于幅值前缀的值再加 1。

[0017] （2）计算第一个码字的长度。第一个码字的长度等于幅值后缀（level_suffix）的长度加上幅值前缀的长度。解码幅值后缀的过程如图 1 所示，幅值后缀的长度由幅值前缀和当前码表号决定。当幅值前缀小于 14 时，幅值后缀的长度等于码表号，当幅值前缀等于 14 时，幅值后缀的长度是 4 比特，当幅值前缀大于 14 时，幅值后缀的长度等于幅值前缀的值减 3。

[0018] （3）计算第一个码字的幅值。第一个码字的幅值可由幅值前缀、幅值后缀和码表号经过逻辑运算得到，具体过程由 H.264 标准规定，如表 2 所示，表中各变量的解释如

下：level 代表当前码字的幅值，level_prefix 是幅值前缀，level_suffix 是幅值后缀，trailingone 代表拖尾 1 的个数，suffixLength 代表当前码表号；>> 和 << 分别代表右移操作和左移操作，% 代表取余数的操作，abs( ) 表示求绝对值的操作，min( ) 表示在两个数中选择较小的一个作为输出，&& 代表逻辑与，|| 代表逻辑或。

[0019] (4) 计算第二个码字的幅值前缀。此步骤同步骤(1)。

[0020] (5)计算第二个码字的长度。由于第二个码字对应的码表号尚未产生，需要同时计算两种可能的长度，计算每种可能长度的具体步骤同步骤(2)；计算第二个码字的幅值，同理，需要同时计算两种可能的幅值，计算每种可能长度的具体步骤同步骤(3)。

[0021] (6) 更新第一个码字输出的码表号。如果码表号等于 6，那么新的码表号为 6；如果码表号是 0，那么新的码表号是 1；否则，新码表号将依照码字幅值的大小来决定是否等于当前码表号还是当前码表号加 1。

[0022] (7) 同时，根据第一个码字输出的新码表号，从步骤(5) 中得到的两个候选结果中选择正确的幅值和码字长度作为第二个非零系数码字的解码结果。

[0023] (8) 更新第二个码字输出的码表号。先计算针对第二个码字的两种可能的新码表号，计算方法同步骤(6)，然后再根据第一个码字输出的码表号，选择正确的码表号作为最终输出。

[0024] 快速解码两个 CALVC 非零系数幅值的整体流程如图 2 所示，经过上述步骤，两个相邻的非零系数幅值的大小和码字长度就可以在一个时钟周期内获得，同时最新的码表号也得到了更新。

[0025] 表 1：幅值前缀的解码码表

幅值前缀的值	视频比特流
0	1
1	01
2	001
3	0001
4	00001
5	000001
6	0000001
7	00000001
8	000000001
9	0000000001
10	00000000001
11	000000000001
12	0000000000001
13	00000000000001
14	000000000000001
15	0000000000000001
…	…

表 2：非零系数幅值的计算方法

level=(Min(15,level_prefix)<<suffixLength)
if(suffixLength>0||level_prefix>=14)
level=level+ level_suffix
if(level_prefix>=15&&suffixLength==0)
level=level+15
if(level_prefix>=16)

```
level=level+ (1<<(level_prefix - 3)) - 4096
if((i==TrailingOnes)&&(TrailingOnes<3))
level=level+2
if(level%2==0)
level=(level+2)>>1
Else
level=( - level - 1)>>1
if(suffixLength==0)
    suffixLength=1
if(Abs(level)>(3<<(suffixLength - 1))&&suffixLength<6)
suffixLength = suffixLength+1
```

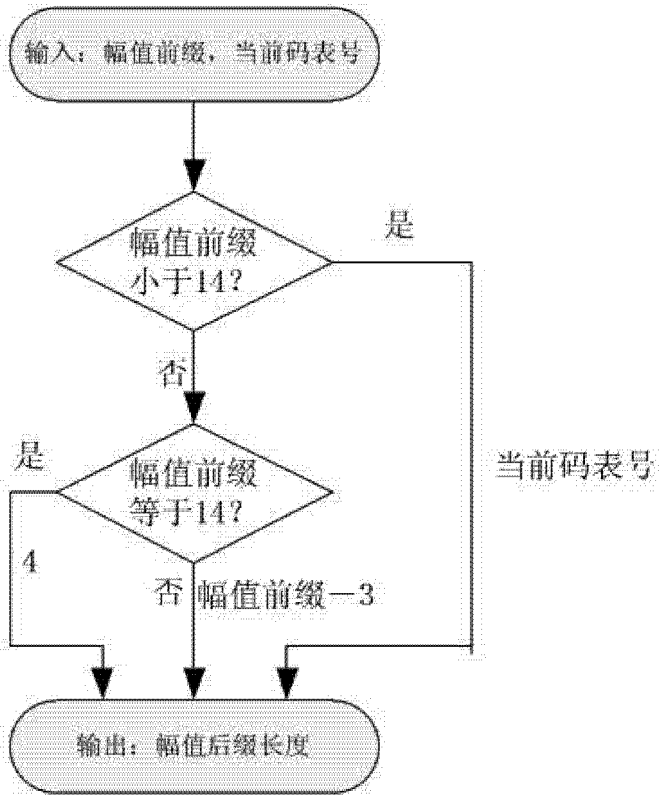

输入：幅值前缀，当前码表号
幅值前缀
小于14?
是
否
是
幅值前缀
等于14?
当前码表号
4
否 幅值前缀－3
输出：幅值后缀长度

图 1

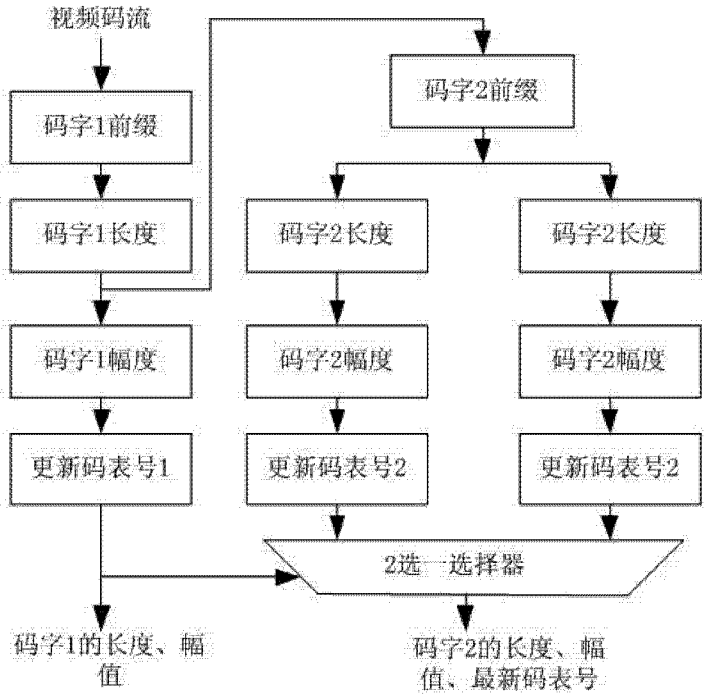

视频码流
码字1前缀
码字1长度
码字1幅度
更新码表号1
码字1的长度、幅值
码字2前缀
码字2长度
码字2幅度
更新码表号2
码字2长度
码字2幅度
更新码表号2
2选一选择器
码字2的长度、幅值、最新码表号

图 2