



[12] 发明专利说明书

[21] ZL 专利号 99801059.6

[45] 授权公告日 2004 年 11 月 10 日

[11] 授权公告号 CN 1175671C

[22] 申请日 1999.4.22 [21] 申请号 99801059.6

[30] 优先权

[32] 1998. 4.30 [33] EP [31] 98401053.8

[86] 国际申请 PCT/IB1999/000731 1999.4.22

[87] 国际公布 WO1999/057673 英 1999.11.11

[85] 进入国家阶段日期 2000.2.29

[71] 专利权人 皇家飞利浦电子有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 N·拜勒尔

审查员 韩 岳

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 程天正 陈景峻

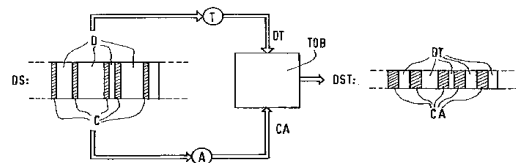
权利要求书 4 页 说明书 17 页 附图 7 页

[54] 发明名称 对数据流进行代码转换的代码转换器和方法

[57] 摘要

数据流(DS)包括时间复接的编码的数据(D)和控制数据(C)。数据流(DS)例如可以是代表图象序列的MPEG类型。对编码的数据(D)进行代码转换(T),从而得到在尺寸上不同于编码的数据(D)的已代码转换的数据(DT)。控制数据(C)被针对已代码转换的数据(DT)进行调整,以便得到在尺寸上几乎与控制数据(C)没有差别的已调整的控制数据(CA)。已代码转换的数据(DT)和已调整的控制数据(CA)被写入到代码转换器输出缓存器(TOB)和从代码转换器输出缓存器(TOB)被读出,以便得到已代码转换的数据流(DST)。这允许有效地使用藉以传输已代码转换的数据流(DST)的传输信道,因而它能给出满意的代码转换质量。根据被包含在代码转换器输出缓存器(TOB)中的数据量和被包含在数据流(DS)中的控制数据(CC),对已代码转换

的数据(DT)的尺寸调整可以进一步给出了满意的代码转换质量。



1. 一种用于对数据流进行代码转换的方法，该数据流包括时间复接的编码的数据和控制数据，其中所述数据流包括控制码，每个控制码规定了对于与它有关的编码数据部分的译码器输入缓存器延时，所述输入缓存器延时是编码数据部分被写入到译码器输入缓存器的第一时刻与所述编码数据部分从所述译码器输入缓存器被读出的第二时刻之间的差值，该方法包括以下步骤：

 对所述编码数据进行代码转换以便得出在大小上不同于所述编码数据的代码转换的数据；

 调整用于所述代码转换的数据的所述控制数据，以便得出自适应的控制数据；

 把所述代码转换的数据和所述自适应的控制数据写入到代码转换器输出缓存器中；

 从所述代码转换器输出缓存器读出所述代码转换的数据和所述自适应的控制数据，以便得出代码转换的数据流；

 计算代码转换器输出缓存器延时，它是代码转换的数据部分被写入到代码转换器输出缓存器的第三时刻与代码转换的数据部分从代码转换器输出缓存器被读出的第四时刻之间的差值；

 计算新的译码器输入缓存器延时，它等于译码器输入缓存器延时和代码转换延时的和值减去代码转换器输出缓存器延时；

 为代码转换的数据部分提供自适应的控制码，它规定了所述新的译码器输入缓存器延时。

2. 如权利要求 1 中所要求的方法，其特征在于，该方法进一步包括以下步骤：

 - 计算代码转换器输出缓存器充满度，它是在代码转换的数据部分被写入到代码转换器输出缓存器的第三时刻被包含在代码转换器输出缓存器中的数据量；

 - 计算译码器输入缓存器充满度，它是在代码转换的数据部分从译码器输入缓存器被读出的第五时刻将要被包含在一个接收代码转换的数据流的译码器输入缓存器中的数据量，译码器输入缓存器充满度是根据代码转换器输出缓存器充满度、对于代码转换的

数据流的投影比特率、以及译码器输入缓存器延时与代码转换延时的和值而被计算的;

- 根据译码器输入缓存器充满度调整量化参量, 以确定将要被包括在代码转换的数据部分中的数据量。

3. 如权利要求 2 中所要求的方法, 其特征在于, 该方法进一步包括以下步骤:

- 确定对于被包括在代码转换的数据部分中的数据量的目标;

- 检验目标是否超过译码器输入缓存器充满度, 如果超过的话, 则减小目标, 以便得到不超过译码器输入缓存器充满度的检验的目标;

- 根据检验的目标调整量化参量。

4. 如权利要求 1 中所要求的方法, 其中在接连的代码转换的数据部分从接收代码转换的数据流的译码器输入缓存器中被读出的时刻之间有一个固定的时间间隔, 所述方法进一步包括以下步骤:

- 确定对于被包括在代码转换的数据部分中的数据量的目标;

- 计算代码转换器输出缓存器充满度, 它是在代码转换的数据部分被写入到代码转换器输出缓存器的第三时刻被包含在代码转换器输出缓存器中的数据量;

- 计算下一个译码器输入缓存器充满度, 它是在下一个代码转换的编码数据部分从译码器输入缓存器被读出之前将要被包含在译码器输入缓存器中的数据量, 下一个译码器输入缓存器充满度是根据目标、代码转换器缓存器充满度、对于代码转换的数据流的投影比特率、以及译码器输入缓存器延时、代码转换延时、与所述固定的时间间隔这三者的和值而被计算的;

- 检验下一个译码器输入缓存器充满度是否超过预定的最大值, 如果超过的话, 则提高目标, 以便得到其下一个译码器输入缓存器充满度不超过预定的最大值的检验的目标;

- 根据检验的目标调整量化参量以确定被包括在代码转换的

数据部分中的数据量。

5. 如权利要求 1 中所要求的方法，其中代码转换的数据部分位于权利要求 1 中提到的代码转换的数据部分的前面，这些数据部分此后分别被称为当前的数据部分和将来的数据部分，该方法进一步包括以下步骤：

- 规定想要的译码器输入缓存器充满度，它是在将来的代码转换的数据部分从译码器输入缓存器被读出时，译码器输入缓存器应当优选包含的数据量；

- 预测与将来的数据部分有关的译码器输入缓存器延时；

- 计算想要的代码转换器输出缓存器充满度，它是在将来的代码转换的数据部分被写入到代码转换器输出缓存器时的第三时刻，代码转换器缓存器应当优选包含的数据量；所述想要的代码转换器输出缓存器充满度是根据对于代码转换的数据流的投影比特率以及预计的译码器输入缓存器延时与代码转换延时的和值而被计算的；

- 根据想要的代码转换器输出缓存器充满度调整量化参量，以确定被包括在当前的代码转换的数据部分中的数据量。

6. 一种用于对数据流进行代码转换的方法，该数据流包括时间复接的编码数据和控制数据，其中所述数据流包括控制码，每个控制码规定了对于与它有关的编码数据部分的译码器输入缓存器延时，所述输入缓存器延时是编码数据部分被写入到译码器输入缓存器的第一时刻与所述编码数据部分从所述译码器输入缓存器被读出的第二时刻之间的差值，所述方法包括以下步骤：

对所述编码数据进行代码转换以便得出在大小上不同于所述编码数据的代码转换的数据；

调整用于所述代码转换的数据的所述控制数据，以便得出自适应的控制数据；

把所述代码转换的数据和所述自适应的控制数据写入到代码转换器输出缓存器中；

从所述代码转换器输出缓存器读出所述代码转换的数据和所述自适应的控制数据，以便得出代码转换的数据流；

根据译码器输入缓存器延时计算量化参量的数值, 所述量化参量确定将要被包括在通过对编码数据部分进行代码转换而得出的代码转换的数据部分中的数据量。

7. 一种用于对数据流进行代码转换的代码转换器, 该代码转换器包括时间复接的编码数据和控制数据, 其中所述数据流包括控制码, 每个控制码规定了对于与它有关的编码数据部分的译码器输入缓存器延时, 所述译码器输入缓存器延时是编码数据部分被写入到译码器输入缓存器的第一时刻与所述编码数据部分从所述译码器输入缓存器被读出的第二时刻之间的差值, 所述代码转换器包括:

代码转换器电路, 用于对所述编码数据进行代码转换以便得出在大小上不同于所述编码数据的代码转换的数据;

控制器, 用于调整所述代码转换的数据的所述控制数据, 以便得出自适应的控制数据, 并且用来根据译码器输入缓存器延时计算量化参量的数值, 所述量化参量确定将要被包括在通过对编码的数据部分进行代码转换而得出的代码转换的数据部分中的数据量;

代码转换器输出缓存器, 用于提供具有代码转换的数据流形式的代码转换的数据和自适应的控制数据。

对数据流进行代码转换的代码转换器和方法

技术领域

- 5 本发明涉及包括编码数据和控制数据的时间复接的数据流的代码转换。数据流例如可以代表按照诸如活动图象专家组 (MPEG) 规定的图象-编码标准而被编码的图象序列。

背景技术

- 10 EP-A 0 690 392 (代理人文档号 PHF 94001) 描述了对与图象序列相对应的已编码数字信号进行代码转换的方法。该方法包括对于与每个当前图象有关的输入数字信号进行译码的步骤。在译码步骤后面是编码步骤。

发明内容

本发明的目的在于提供能获得满意质量的数据流的代码转换。

- 15 本发明考虑了下列方面。数据流可以不仅包括时间复接编码的数据还包括控制数据。在代码转换中, 将把数据流分割成编码的子数据流和控制数据流。该编码的子数据流通过代码转换步骤而代码转换成被代码转换的子数据流。可能必须通过自适应步骤调整用于被代码转换的数据的控制数据。在这种情况下, 该控制子数据流被有效地转换成自适
- 20 应的子控制数据流。被代码转换的子数据流和自适应的子控制数据流可被组合, 以得出被代码转换的数据流, 它是原先被包含在编码的数据流中的数据的不同的代表。

- 某些编码标准, 例如 MPEG 图象-编码标准, 不固定编码的数据的大小, 但隐含地固定控制数据的大小。在这样的编码标准中, 编码的数据可被译码和再编码, 以便得出被代码转换的数据, 它在大小上不同
- 25 于编码的数据。然而, 不可能也对于控制数据这样处理而不背离编码标准。如果数据流必须被进行代码转换而同时保持它遵从其编码标准, 则可能要修改控制数据的内容而不是其大小。结果, 被代码转换的子数据流和自适应的子控制数据流将具有不同的比特率。

- 30 按照本发明, 被代码转换的数据和自适应的控制数据通过写入步骤将被写入到译码输出缓存器, 以及通过读出步骤从译码输出缓存器读出, 以便得到被代码转换的数据流。因之, 在代码转换的子数据流和

子控制数据流之间的任一比特率差异将被平均化,以便代码转换的子数据流具有基本上恒定的比特率,或至少,它基本上不取决于输送的是编码的数据还是控制数据。因此,本发明使得代码转换的数据流的比特率能连续地等于一个确定的传输信道可处理的最大比特率,或至少相当接近于这个最大比特率。因之,由于本发明允许有效地使用一个藉以发送代码转换的数据流的传输信道,它使得代码转换的数据具有相当大的大小。因此,本发明确保在代码转换时保存最大的数据量。从而,本发明允许进行具有满意的质量的代码转换。

本发明的技术方案包括:

10 (一)一种用于对数据流进行代码转换的方法,该数据流包括时间复接的编码的数据和控制数据,其中所述数据流包括控制码,每个控制码规定了对于与它有关的编码数据部分的译码器输入缓存器延时,所述输入缓存器延时是编码数据部分被写入到译码器输入缓存器的第一时刻与所述编码数据部分从所述译码器输入缓存器被读出的第
15 二时刻之间的差值,该方法包括以下步骤:

对所述编码数据进行代码转换以便得出在大小上不同于所述编码数据的代码转换的数据;

调整用于所述代码转换的数据的所述控制数据,以便得出自适应的控制数据;

20 把所述代码转换的数据和所述自适应的控制数据写入到代码转换器输出缓存器中;

从所述代码转换器输出缓存器读出所述代码转换的数据和所述自适应的控制数据,以便得出代码转换的数据流;

25 计算代码转换器输出缓存器延时,它是代码转换的数据部分被写入到代码转换器输出缓存器的第三时刻与代码转换的数据部分从代码转换器输出缓存器被读出的第四时刻之间的差值;

计算新的译码器输入缓存器延时,它等于译码器输入缓存器延时和代码转换延时的和值减去代码转换器输出缓存器延时;

30 为代码转换的数据部分提供自适应的控制码,它规定了所述新的译码器输入缓存器延时。

(二)一种用于对数据流进行代码转换的方法,该数据流包括时间复接的编码数据和控制数据,其中所述数据流包括控制码,每个控制码规定了对于与它有关的编码数据部分的译码器输入缓存器延时,

所述输入缓存器延时是编码数据部分被写入到译码器输入缓存器的第一时刻与所述编码数据部分从所述译码器输入缓存器被读出的第二时刻之间的差值，所述方法包括以下步骤：

5 对所述编码数据进行代码转换以便得出在大小上不同于所述编码数据的代码转换的数据；

调整用于所述代码转换的数据的所述控制数据，以便得出自适应的控制数据；

把所述代码转换的数据和所述自适应的控制数据写入到代码转换器输出缓存器中；

10 从所述代码转换器输出缓存器读出所述代码转换的数据和所述自适应的控制数据，以便得出代码转换的数据流；

根据译码器输入缓存器延时计算量化参量的数值，所述量化参量确定将要被包括在通过对编码数据部分进行代码转换而得出的代码转换的数据部分中的数据量。

15 (三) 一种用于对数据流进行代码转换的代码转换器，该代码转换器包括时间复接的编码数据和控制数据，其中所述数据流包括控制码，每个控制码规定了对于与它有关的编码数据部分的译码器输入缓存器延时，所述译码器输入缓存器延时是编码数据部分被写入到译码器输入缓存器的第一时刻与所述编码数据部分从所述译码器输入缓存器被读出的第二时刻之间的差值，所述代码转换器包括：

20 代码转换器电路，用于对所述编码数据进行代码转换以便得出在大小上不同于所述编码数据的代码转换的数据；

25 控制器，用于调整所述代码转换的数据的所述控制数据，以便得出自适应的控制数据，并且用来根据译码器输入缓存器延时计算量化参量的数值，所述量化参量确定将要被包括在通过对编码的数据部分进行代码转换而得出的代码转换的数据部分中的数据量；

代码转换器输出缓存器，用于提供具有代码转换的数据流形式的代码转换的数据和自适应的控制数据。

30 从后面所描述的附图以及将参照附图进行的描述中，将可以明了本发明以及可任选地用来实施本发明的附加的特性。

附图说明

图 1 是说明如本发明技术方案（一）代表的本发明的基本特性的概念性图。

图 2 到 6 分别是说明如本发明的其他技术方案代表的本发明的基

本特性的概念性图。

图 7 是按照本发明的代码转换器的实例的方框图。

具体实施方式

首先，将就参考符号的使用作出某些说明。在所有图上，用相同的字母代码表示相同的项目。在单个图上，可能显示各个相同的项目。在这种情况下，在字母代码上加上号码，以便互相区分相同的项目。如果相同的项目的号码是运行的参量，则号码将放在括号之间。在说明书和权利要求中，参考符号中的任何数字可以被省略，如果这是合适的话。

图 1 显示了本发明的基本特性。要被代码转换的数据流 DS 包括时间复接的编码数据 D 和控制数据 C。通过代码转换步骤，编码的数据 D 被进行代码转换 (T)，以便得出代码转换的数据 DT。代码转换的数据 DT 在大小上不同于编码的数据 D。通过自适应步骤，控制数据 C 被针对代码转换的数据 DT 进行调整 (A)，这样可得出自适应的控制数据 CA。自适应的控制数据 CA 和控制数据 C 在大小上基本上是相等的。通过写入步骤，代码转换的数据 DT 和自适应的控制数据 CA 被写入代码转换器输出缓存器 TOB，并且通过读出步骤将其从代码转换器输出缓存器 TOB 读出，以便得出代码转换的数据流 DST。

图 2 显示除了参照图 1 所讨论的特性以外的特性。数据流 DS 包括控制码 CC。控制码规定了对于与其有关的编码的数据部分 P 的译码器输入缓存器延时 ΔT_{dib} 。译码器输入缓存器延时 ΔT_{dib} 是编码数据部分 P 被写入译码器输入缓存器 DIB1 的时刻 T_{wd1} 与编码数据部分 P 从译码器输入缓存器 DIB1 被读出的时刻 T_{rd1} 之间的差值。以步长 $S/\Delta T_{tob}$ ，计算代码转换器输出缓存器延时 ΔT_{tob} 。代码转换器输出缓存器延时 ΔT_{tob} 是通过将编码的数据部分 P 进行代码转换而得到的代码转换的数据部分 PT 被写入代码转换器输出缓存器 TOB 的时刻 T_{wt} 、与被代码转换的数据部分 PT 从代码转换器输出缓存器 TOB 被读出的时刻 T_{rt} 之间的差值。按步长 $S/\Delta T_{dib_new}$ ，计算新的译码器输入缓存器延时 ΔT_{dib_new} 。新的译码器输入缓存器延时 ΔT_{dib_new} 基本上等于译码器输入缓存器延时 ΔT_{dib} 与代码转换延时 ΔT_{trans} 的和值减去代码转换器输出缓存器延时 ΔT_{tob} 。按步长 S/CCA ，提供对于代码转换的数据部分 PT 的自适应的控制码 CCA，自适应的控制码 CCA 规定了新的译码器输入缓存器延时 ΔT_{dib_new} 。

对于图 2 所示的特性，考虑以下的方面。控制码 CC 有效地确定时间

图案, 数据部分 P 将按照这个时间图案来从接收数据流 DS 的译码端处的译码器输入缓存器 DIB1 中读出。在许多应用项中, 希望按照相同的时间图案在接收代码转换的数据流 DST 的译码端处从译码器输入缓存器 DIB2 中读出数据部分 P。

- 5 原则上, 有可能只把控制码 CC 从数据流 DS 复制成代码转换的数据流 DST。也就是, 被使用于数据部分 P 的译码器输入缓存器延时 ΔT_{dib} 也可被使用于通过对于数据部分 P 进行代码转换而得到的代码转换的数据部分 PT。然而, 如果这样做, 则只要以下的条件满足, 代码转换的数据部分 PT 就将按照与编码的数据部分 P 相同的时间图案被读出。在代码转换的数据流 DST 中, 代码转换的数据部分 PT 应当相互地以与数据流 DS 中的编码的数据部分 P 完全相同的方式被放置。这是指, 每个代码转换的数据部分 PT 应当具有特定的大小。在代码转换的过程中, 要确保每个代码转换的数据部分 PT 具有所需要的大小可能是相当困难, 因而其代价是很昂贵的, 或可能甚至不能保证做到这一点。而且, 为了满足这个条件, 有可能一个或多个代码转换的数据部分应当具有相当小的大小。因此, 在实际代码转换以便得到这种小大小代码转换的数据部分的过程中, 可能丢失相当大量的信息。这可能对通过译码代码转换的数据流 DST 而得到的数据的质量产生有害影响。
- 10
- 15

- 图 2 所示的特性提供以下有利的效果。新的译码器输入缓存器延时 ΔT_{dib_new} 是代码转换的数据部分 PT 被写入译码器输入缓存器 DIB2 的时刻 T_{wd2} 与代码转换的数据部分从译码器输入缓存器 DIB2 被读出的时刻 T_{rd2} 之间的差值。假定译码器输入缓存器 DIB2 被直接耦合到代码转换器输出缓存器 TOB。在这种情况下, 编码的数据部分 P 被写入代码转换器输出缓存器 TOB 的时刻与代码转换的数据部分 PT 从译码器输入缓存器 DIB 被读出的时刻之间的差值基本上等于译码器输入缓冲器延时 ΔT_{dib} 与固定的代码转换延时 ΔT_{trans} 之和值。因此, 代码转换的数据部分 PT 从译码器输入存储器 DIB2 被读出的时刻与编码的数据部分 P 从译码器输入存储器 DIB1 被读出的时刻之间的差值基本上等于固定的代码转换延时 ΔT_{trans} 。这样, 代码转换的数据部分 PT 按照与编码的数据部分 P 从译码器输入缓存器 DIB1 被读出的相同的时间图案从译码器输入缓存器 DIB2 被读出。如果代码转换的数据部分 PT 在从代码转换器输出缓存器 TOB 被读出后经一定的传输延时 ΔT_{td} 到达译码器输入缓存器 DIB, 则这同样地适
- 20
- 25
- 30

用。因此，图 2 所示的特性允许代码转换的数据部分 PT 按照想要的时间图案被读出而不在大小上去限制被代码转换的数据部分 PT，这将需要相当高的花费以及影响质量，正如前面所解释的那样。因此，图 2 所示的特性给出了相当经济的实施方案和满意的质量。

5 关于图 2 所示的特性，将作以下的附加说明。代码转换器输出延时 ΔT_{tob} 可以以各种不同的方式来计算。例如，它可以根据以下的参量进行计算：(a) 当代码转换的数据部分 PT 被写入这个缓存器时，在代码转换器输出缓存器 TOB 中所包含的数据量，以及 (b) 将从这个缓存器读出数据的速率。被包含在代码转换器缓存器中的数据量可通过对被写入到代码
10 转换器输出缓存器 TOB 和从该缓存器被读出的数据量进行计数而被确定。将从代码转换器输出缓存器 TOB 读出数据的速率是代码转换的数据流 DST 的比特速率，它可以由控制器来确定。下面是计算某个代码转换的数据部分 PT 的代码转换器输出延时 ΔT_{tob} 的例子。假定代码转换输出缓存器 TOB 在代码转换的数据部分 PT 被写入到代码转换器输出缓存器 TOB 的时刻包
15 含 500 kbits，以及代码转换的数据流 DST 的比特速率是 5Mbits/秒。那么，可以容易地计算出：被代码转换的数据部分将在 100ms 的延时后从代码转换器输出缓存器 TOB 被读出，因此，该数据部分的代码转换器输出缓存器延时 ΔT_{tob} 是 100 毫秒。然而，代码转换器输出延时 ΔT_{tob} 也可以通过测
20 量代码转换的数据部分 PT 被写入到代码转换器输出缓存器 TOB 的时刻与代码转换的数据部分从这个缓存器被读出的时刻而被计算。在这种情况下，代码转换的数据流 DST 在代码转换结束时可以附加地被延时，以便使得控制码 CC 以反作用的方式被调整。

25 固定的代码转换延时 ΔT_{trans} 最好应当等于或大于由代码转换过程实际引入的延时。固定的代码转换延时 ΔT_{trans} 涉及图象质量与发送端与接收端间的总的延时之间的折衷。也就是，如果固定的代码转换延时 ΔT_{trans} 具有相当大的数值，则这将对图象质量起作用。然而，如果例如有足球比赛的实时传输，则相当大的该数值可能是不满意的。

30 图 3、4、5、和 6 显示了另一些可任选地被用来实施本发明的附加特性。这些附图中的每个包括一个图形，其水平轴代表时间 T 和垂直轴代表数据量 X。每个图形包括三条曲线 A、B、和 C。曲线 A 代表作为时间的函数的被写入到代码转换器输出缓存器 TOB 的数据量。曲线 B 代表从译码器输出缓存器 TOB 移出的以及随后在译码端被写入到译码器输入缓存器

DIB 的数据量。曲线 C 代表作为时间的函数的从译码器输入缓存器 DIB 被读出的数据量。代码转换输出缓存器 TOB 包含的数据量在任何时刻都是在这个时刻的曲线 A 和 B 之间的差值。译码器输入缓存器 DIB 包含的数据量在任何时刻都是在这个时刻的曲线 B 和 C 之间的差值。

- 5 图 3 显示了除了参照图 2 描述的特性以外的以下的特性。按步长 S/F_{tob} , 确定代码转换器输出缓存器充满度 F_{tob} 。代码转换器输出缓存器充满度 F_{tob} 是在代码转换的数据部分 $PT(i)$ 被写入到代码转换器输出缓存器 TOB 的时刻 T_{wt} 代码转换器输出缓存器 TOB 中所包含的数据量 $A-B$ 。按步长 S/F_{dib} , 计算译码器输入缓存器充满度 F_{dib} 。译码器输入缓存器充满度 F_{dib} 是在代码转换的数据部分 $PT(i)$ 从译码器输入缓存器 DIB 被读出的时刻 T_{rd} 在接收代码转换的数据流 DST 的译码器输入缓存器中所包含的数据量 $B-C$ 。译码器输入缓存器充满度 F_{dib} 是根据代码转换器输出缓存器充满度 F_{tob} 、代码转换的数据流 DST 的投影比特速率 R 、以及译码器输入缓存器延时 ΔT_{dib} 与代码转换延时 ΔT_{trans} 的和值而被确定的。按步长 $S/QP-F_{dib}$, 确定数据量 $XPT(i)$ (它将被包括在代码转换的数据部分 PT 中) 的量化参量 QP 根据译码器输入缓存器充满度 F_{dib} 而被调整。
- 10
- 15

- 关于图 3 所示的特性, 考虑以下的方面。如果在某个时间间隔内从译码器输入缓存器 DIB 读出的数据量超过被写入到这个缓存器的数据量, 则这个缓存器将出现下溢的风险。也就是, 在要从译码器输入缓存器 DIB 读出一定的代码转换的数据部分的时刻, 代码转换的数据部分还没有完全地被写入到这个缓存器。反之, 如果在某个时间间隔内被写入到译码器输入缓存器 DIB 的数据量超过从这个缓存器读出的数据量, 则这个缓存器将出现上溢的风险。也就是, 在一定的代码转换的数据部分到达译码器输入缓存器 DIB 的时刻, 在这个缓存器 DIB 中没有剩下的足够的空间用来存储代码转换的数据部分。在任何情况下, 下溢或上溢都将引起可供译码处理用的代码转换的数据的一定程度的丢失。这在译码端会引起恼人的结果。
- 20
- 25

- 图 3 所示的特性提供以下的有利的结果。包含在代码转换的数据部分 $PT(i)$ 中的数据量 $XPT(i)$ 将取决于译码器缓存器充满度 F_{dib} 。也就是, 包含在代码转换的数据部分 $PT(i)$ 中的数据量 $XPT(i)$ 将取决于在代码转换的数据部分 $PT(i)$ 从译码器输入缓存器 DIB 被读出的时刻 T_{rd} 在这个缓存器中包含的数据量。这防止译码器输入缓冲器 DIB 的下溢或上
- 30

溢。例如,假定计算出译码器输入缓存器 DIB 是相当满的。在这种情况下,量化参量 QP 可被设定为这样的值,以使得代码转换的数据部分 PT (i) 包括相当大的数据量 XPT (i),以避免溢出。反之,如果计算出译码器输入缓存器 DIB 是相当空的,则量化参量 QP 可被设定为这样的值,以使得
5 代码转换的数据部分 PT 包括相当小的数据量 XPT (i),以避免下溢。因此,由于图 3 所示的特性使得能对抗下溢和上溢,所以这些特性提供了在译码端处得到的满意的质量。

关于图 3 所示的特性,作出以下的附加说明。译码器输入缓存器充满度 Fdib 可以以各种不同的方式被计算。例如,可以通过首先计算哪些
10 数据量 DX 将在一个开始于代码转换的数据部分 PT 被写入到代码转换器输出缓存器的时刻 Twt 以及具有等于译码器输入缓存器延时 ΔT_{dib} 与代码转换延时 ΔT_{trans} 之和的长度的时间间隔里从代码转换器输出缓存器 TOB 转移到译码器输入缓存器 DIB,从而进行所述的计算。图 4 所包括的图形显示可以应用以下的法则:数据量 DX 等于代码转换器输入缓存器充满度
15 Ftob 与译码器输入缓存器充满度 Fdib 之和。这样,译码器输入缓存器充满度 Fdib 是数据量 DX 与代码转换器输入缓存器充满度 Ftob 之间的差值。计算译码器输入缓存器充满度 Fdib 的另一种方式是利用如参照图 2 进行解释所计算的新的译码器输入缓存器延时 ΔT_{dib_new} 。译码器输入缓存器充满度 Fdib 是在一个起始于代码转换的数据部分 PT 将从代码转换器输出
20 缓存器 TOB 被读出的时刻 Trt 以及具有等于新的译码器输入缓存器延时 ΔT_{dib_new} 的长度的时间间隔内从代码转换器输出缓存器 TOB 转移到译码器输入缓存器 DIB。

图 4 显示了除了参照图 3 讨论的特性以外的以下的特性。按步长 S/Ttarget, 针对被包括在代码转换的数据部分 PT(i) 中的数据量 XPT(i) 来确定目标 Ttarget。按步长 S/Ttarget-ver, 检验目标 Ttarget 是否超
25 过译码器输入缓存器充满度 Fdib。如果超过,则目标 Ttarget 被减小,以便得出检验的目标 Ttarget-ver 不超过译码器输入缓存器充满度 Fdib。按步长 S/QP-Ttarget-ver, 根据被检验的目标 Ttarget-ver 来调整量化参量 QP。

30 关于图 4 所示的特性,考虑以下的方面。首先,关于下溢,应当考虑以下的情形。在译码端,代码转换的数据部分 PT (i) 在此前讨论的新的译码器输入缓存器延时 ΔT_{dib_new} 所确定的时刻被立即从译码器输入

缓存器 DIB 移出。如果 T_{target} 超过译码器输入缓存器充满度 F_{dib} , 则这会导致下溢出。也就是, 会出现应当存在于译码器输入缓存器 DIB 中的代码转换的数据部分 $PT(i)$ 只是部分地存在或在最坏的情况下全部不存在的风 险。

- 5 其次, 对于调整量化参量 QP , 还会有除了译码器输入缓存器充满度 F_{dib} 以外的准则。例如, 要被代码转换的数据流可以包括大小上相对较小的编码的数据部分和大小上相对较大的数据部分。通常, 如果相对较大的编码的数据部分也被代码转换成相对较大的代码转换的数据部分, 则将得出满意的代码转换质量。相对较小的编码的数据部分优选地被代码转换
- 10 成相对较小的代码转换的数据部分, 以便于有效地留下供相对较大的代码转换的数据部分用的足够的空间。这样, 为了得到满意的代码转换的结果, 可以希望根据要被代码转换的被包含在编码的数据部分中的数据量来调整量化参量 QP 。

- 图 4 所示的特性提供以下的有利的结果。量化参量 QP 只在目标
- 15 T_{target} 超过译码器输入缓存器充满度 F_{dib} , 才直接受译码器输入缓存器充满度 F_{dib} 影响。也就是, 量化参量 QP 只在有严重的下溢的风险时, 才直接受译码器输入缓存器充满度 F_{dib} 影响。无论如何, 如果没有下溢的风险, 量化参量 QP 就根据目标 T_{target} 来调整, 正如原先规定的。目标 T_{target} 可以这样地定义, 以使得得出满意的代码转换质量。例如, 它可
- 20 以根据编码的数据部分的大小来定义, 正如此前所讨论的。目标 T_{target} 的检验实际上是一种防护, 原则上, 它不应当太多地进行。这样, 通过能提供主要根据除译码器输入缓存器充满度 F_{dib} 以外的一个或多个参量来调整量化参量 QP 、并且同时提供对于下溢的防护, 从而使图 4 所示的特性提供满意的代码转换质量。

- 25 图 5 显示了除了参照图 2 讨论的特性以外的以下的特性。在接连的代码转换的数据部分 PT 要从接收代码转换的数据流的译码器输入缓存器被读出的时刻之间具有固定的时间间隔 ΔT_{pt} 。按步长 S/T_{target} , 对于被包括在代码转换的数据部分 $PT(i)$ 中的数据量 $X_{PT}(i)$ 确定目标 T_{target} 。按步长 S/F_{tob} , 确定代码转换器输出缓存器充满度 F_{tob} 。代码转换器输出
- 30 缓存器充满度 F_{tob} 是在代码转换的数据部分 $PT(i)$ 被写入到代码转换器输出缓存器 TOB 的时刻 $T_{wt}(i)$ 代码转换器输出缓存器 TOB 中所包含的数据量 $A-B$ 。按步长 S/F_{dib_next} , 计算随后的译码器输入缓存器充满

度 F_{dib_next} 。下一个译码器输入缓存器充满度 F_{dib} 是刚好在随后的代码转换的数据部分 $PT(i+1)$ 从译码器输入缓存器 DIB 被读出之前的时刻 $Trd(i+1)-\delta$ 将要在译码器输入缓存器 DIB 中包含的数据量 $B-C$ 。下一个译码器输入缓存器充满度 F_{dib_next} 是根据代码转换器输出缓存器充满度 F_{tob} 、代码转换的数据流的投影比特速率 R 、以及译码器输入缓存器延时 ΔT_{dib} 、代码转换延时 ΔT_{trans} 、与固定的时间间隔 ΔT_{pt} 这三者的和值而被计算的。按步长 S/T_{target_ver} ，检验下一个译码器输入缓存器充满度 F_{dib_next} 是否超过预定的最大值 MAX 。如果超过，则目标 T_{target} 被增加，以便得出检验的目标 T_{target_ver} ，对于它，下一个译码器输入缓存器充满度 F_{dib_next} 不超过预定的最大值 MAX 。按步长 $S/QP-T_{target_ver}$ ，根据被检验的目标 T_{target_ver} 调整量化参量 QP 。

关于图 5 所示的特性，考虑类似于关于图 3 和 4 所示的特性的那些方面的以下的方面。其它考虑是如下。代码转换的数据部分 $PT(i)$ 可包含一个小于在时刻 $Trd(i)$ 和 $Trd(i+1)-\delta$ 之间的固定的时间间隔 ΔT_{pt} 期间将被写入到译码器输入缓存器 DIB 的数据量的数据量 $XPT(i)$ 。在这种情况下，被包含在译码器输入缓存器 DIB 中的数据量在该时间间隔期间将增大。如果译码器输入缓存器 DIB 已经被填充以相当大量的数据量以及如果代码转换的数据部分 $PT(i)$ 包含相对较小量的数据量 $XPT(i)$ ，则可能出现上溢。

图 5 所示的特性提供以下的有利的结果。下一个译码器缓存器充满度 F_{dib_next} 实际上是在刚好在下一个代码转换的数据部分 $PT(i+1)$ 从译码器输出缓存器被读出以前的时刻 $Trd(i+1)-\delta$ 将要被包括在译码器输入缓存器 DIB 中的数据量的预计值。这个预计值是根据初始地被确定的目标 T_{target} 而作出的。检验在时刻 $Trd(i+1)-\delta$ 在译码器输入缓存器 DIB 中的预计的数据量是否低于想要的最大值 MAX 。也就是，检验对于初始地确定的目标 T_{target} 是否多半出现还是不出现上溢。如果多半出现上溢，则目标 T_{target} 充分地加大，以使得多半不再出现上溢。这样，图 5 所示的特性提供了对译码器输入缓存器上溢的防护，否则可能发生上溢。有关这些特性的其它优点类似于此前对于图 3 和 4 所示的特性所提到的优点。

图 6 显示了除了参照图 2 讨论的特性以外的以下的特性。代码转换的数据部分 $PT(i-k)$ 是位处在参照图 2 讨论的代码转换的数据部分 $PT(i)$ 的前面。这些数据部分在以后分别被称为当前的代码转换的数据部分

PT(i-k)和将来的代码转换的数据部分 PT(i)。按步长 S/F_{dib_des} , 定义了想要的译码器输入缓存器充满度 F_{dib_des} 。想要的译码器输入缓存器充满度 F_{dib_des} 是在将来的代码转换的数据部分 PT(i) 从译码器输入缓存器 DIB 被读出的时刻 $T_{rd}(i)$ 译码器输入缓存器 DIB 应当优选地包含的数据量 B-C。按步长 $S/\Delta T_{dib}(i)$, 与未来数据部分 (Pi) 相关的译码器输入缓冲器延时 $\Delta T_{dib}(i)$ 被进行预计。按步长 S/F_{tob_des} , 计算想要的代码转换器输出缓存器充满度 F_{tob_des} 。想要的代码转换器输出缓存器充满度 F_{tob_des} 是在将来的代码转换的数据部分被写入到代码转换器输出缓存器 TOB 的时刻 $T_{wt}(i)$ 代码转换器输出缓存器 TOB 应当优选地包含的数据量 A-B。想要的代码转换器输出缓存器充满度是根据代码转换数据流的投影比特速率 R 和预计的译码器输入缓存器延时 $\Delta T_{dib}(i)$ 而被计算的。按步长 $S/QP-F_{tob_des}$, 根据想要的代码转换器输出缓存器充满度 F_{tob_des} 来调整用于确定被包括在当前的代码转换的数据部分 PT(i-k) 中的数据量 $XPT(i-k)$ 的量化参量 QP。

关于图 6 所示的特性, 考虑以下的方面。已经提到, 如果一个包括相对较大的数据量的编码的数据部分被代码转换为一个也包括相对较大的数据量的代码转换的数据部分, 则可以得到满意的代码转换质量。然而, 正如对于图 4 所示的特性所讨论的, 当一个相对较大的代码转换的数据部分 PT 从译码器输入缓存器 DIB 被读出时有下溢的风险。也就是, 在这个关键时刻存在有相对较大的代码转换的数据部分还没有完全被写入到译码器输入缓存器 DIB 的风险。如果应用图 4 所示的特性, 也就是, 如果试图保持在相对较大的代码转换的数据部分中的数据量低于可能发生下溢的临界值, 则这个风险得以减小。在这种情况下, 避免下溢是使得该数据量被包括在相对较大的代码转换的数据部分中的准则, 而优选地可以应用一个或多个其它的准则, 例如, 一个藉以得出所讨论的代码转换的数据部分的编码的数据部分中所包含的数据量。因此希望被包含在译码器输入缓存器 DIB 中的数据量保持在一定的极限内, 而不用像图 4 所示的特性那样采取任何的防护, 或至少是如果有防护的话, 也不太经常地采取行动。

图 6 所示的特性提供以下的有利的结果。量化参量 QP、以及因此被包括在当前的代码转换的数据部分 PK(i-k) 中的数据量 $XPT(i-k)$ 被这样地调整, 以使得对于将来的代码转换的数据部分 PT(i) 的被包含在译码器输入缓存器中的数据量处在想要的水平, 或至少足够地接近它。因此, 图

6 所示的特性提供一种长期策略：当将来的代码转换的数据部分 PT(i) 要被代码转换时，它防止必须采取短期测量来避免译码器输入缓存器 DIB 中的下溢。将来的代码转换的数据部分 PT(i) 例如可以是已知或预期包括相对较大的数据量的类型。在这种情况下，想要的译码器输入缓存器充满度 Fdib_des 优选地是相当高的，以便使得将来的代码转换的数据部分 PT(i) 包含相对较大的数据量。因此，这减小了使任何防护（如图 4 所示的特性）起作用的机会。图 6 所示的特性从而提供了能给出满意的代码转换质量的预期调整量化参量 QP 的要素。

图 7 显示了按照本发明的代码转换器的例子，在其中可以单独地或者组合地实施此前参照图 1 到 6 所讨论的任何的特性。被图 7 的代码转换器接收的数据流 DS 可以是代表图象序列的 MPEG 类型，在其中每个编码的数据部分 P 代表不同的图象。代码转换的数据流 DST 可以同样地是 MPEG 类型，在其中每个代码转换的数据部分 PT 代表与藉以得出它的编码的数据部分 P 相同的图象。代码转换的数据流 DST 具有比数据流 DS 低的比特率。在这种情况下，图 7 的代码转换器实现行比特率变换。

图 7 的代码转换器包括此前提到的代码转换器输出缓存器存储器 TOB，以及另外还有代码转换电路 TRC 和控制器 CON。代码转换电路 TRC 包括可变长度译码器 VLD、可变长度编码器 VLC、量化器 Q、两个去量化器 DQ1 和 DQP、离散余弦变换器 DCT、离散反余弦变换器 IDCT、存储器 MEM、运动补偿器 MC、和两个减法器 S1 和 S2。在 EP-A 0 690 392 中更详细地描述了与代码转换电路 TRC 相同的代码转换电路。

代码转换电路 TRC 把编码的数据部分 P 代码转换为代码转换的数据部分 PT。被包括在代码转换的数据部分中的数据量由量化器 Q 的量化参量 QP 确定。编码的数据部分 P 和代码转换的数据部分 PT 都包括量化的 DCT 系数。量化器 QP 在实际上去量化 DCT 系数，该系数首先在编码时被量化，然后被去量化器 DQ1 去量化。用某个量化参量值 QP 对 DCT 系数的量化意味着：DCT 系数的数值被除以量化参量值 QP，然后被四舍五入到最接近的可能的整数。例如，假定 DCT 系数可以是在 -128 与 127 之间的整数值。在这种情况下，需要 8 比特来以二进制表示 DCT 系数。如果量化器 Q 用量化参量 4 来量化 DCT 系数，则将得到量化的 DCT 系数，它可以是在 -32 与 31 之间的任何整数值。在这种情况下，只需要 6 比特来表示量化的 DCT 系数。然而，如果量化器 Q 用量化参量 8 来量化 DCT 系数，

则将得到一个量化的 DCT 系数,它可以是在-16 与 15 之间的任何整数值。在这种情况下,只需要 5 比特来表示量化的 DCT 系数。

控制器 CON 实际上实行对于图 1 到 6 所示的任何特性所描述的步骤。为此,控制器接收来自编码的数据流 DS 的参量,例如在编码的数据流 DS 中的控制码 CC 规定的译码器输入缓存器延时 ΔT_{dib} 。应当指出,控制码 CC 相应于 MPEG 标准中的所谓的“vbv 缓存器延时”。控制器 CON 也可接收来自代码转换器输出缓存器 TOB 的有关数据量的一个或多个参量,它被包括在这个缓存器或代码转换器输出缓存器延时 ΔT_{tob} 中或这二者中。而且,控制器 CON 可以确定或得知代码转换的数据流 DST 的投影比特率 R。根据控制器 CON 接收的或在控制器中可提供的这样的参量,它计算对于代码转换的数据部分 PT 的行动译码器输入缓存器延时 ΔT_{dib_new} 。控制器 CON 也可根据按照对于图 3 到 6 描述的特性的一个或多个算法来调整量化参量。控制器 CON 可以具有适当地编程的计算机的形式。

控制器 CON 可以有各种不同的藉以调整量化参量 QP 的方式。现在将描述一个例子。对于一个已经被代码转换的图象,控制器 CON 计算 (1) 它采用的量化参量 QP 和 (2) 已经得出的代码转换的数据量 (例如,以输出比特数计) 的乘积。这个乘积此后被称为图象的复杂度。复杂度可以说是编码该图象如何困难的量度。如果量化参量 QP 具有相对较大的数值,则可以认为该图象很难编码,这是指,使用相对较少的比特来表示 DCT 系数,以及相当多的 DCT 系数应当被四舍五入为“0”,但无论如何,已经得出相对较大的代码转换的数据量。在这种情况下,作为量化参量 QP 与已经得出的数据量的乘积的复杂度将具有相对较高的数值,从而表示图象很难编码。

假定某种编码类型 I、P、或 B 的图象要被代码转换。在代码转换该图象时采用的量化参量 QP 的基本值被计算如下。基本值等于相同类型 I、P、或 B 的最后的代码转换的图象的复杂度分别除以图象目标。图象目标是应当理想地通过代码转换该图象而得到的数据量。它可以以各种不同的方式被规定。此后将给出两个例子,一个是短期方法,另一个是长期方法。应当指出,量化参量 QP 的基本值实际上是根据先前的编码经验被确定的。假定两个接连的图象的各自的复杂性相差不多。这个假定在大多数情况下是正确的。

用来计算量化参量 QP 的基本值的图象目标可被规定如下。计算了刚

好在图象的代码转换的数据从译码器输入缓存器被读出之前的时刻的译码器输入缓存器充满度。参照图 3, 通过表示图象的代码转换的数据的 $PT(i)$, 根据在 Twt 和 Trd 之间的时间范围内的代码转换器输出缓存器充满度 $Ftob$ 和投影比特率 R , 可以计算译码器输入缓存器充满度 $Fdib$, 这个时间范围的长度是 $\Delta Tdib + \Delta Ttrans$ 。规定了刚好在图象的代码转换的数据从译码器输入缓存器被读出之后的译码器输入缓存器充满度应该是多少。参照图 3, 规定了刚好在时刻 Trd 后在曲线 B 和 C 之间的差值应该是多少, 这个差值是想要的译码器输入缓存器充满度。图象目标然后等于译码器缓存器充满度 $Fdib$ 与想要的译码器缓存器充满度的差值。这样的确定图象目标的方式可被描述如下。在图象被代码转换以前, 根据译码器输入缓存器充满度确定“我们在哪里?”和“我们想要在哪里?”。这二者之间的差别就是图象目标。

有可能对于每个图象规定固定的想要的译码器输入缓存器充满度, 由此可以作出编码类型 I、P、和 B 之间的区别, 以及计算此前描述的图象目标。这可被看作为使用短期目标的方法, 短期相应于一个图象的持续时间。然而, 如果使用根据长期目标的方法, 则通常将得到更好的图象质量, 长期相应于 N 个图象的序列, N 是整数。这在可以以序列中的其它图象的图象目标为代价, 把较高的图象目标分配给 N 个图象的序列中的一个相对较复杂的图象的意义上, 能提供图象目标方面的灵活性。因此, 可以得到甚至更高的图象质量。

以下是使用长期目标的方法的例子。对于将来的时刻规定了想要的译码器输入缓存器充满度。将来的时刻是刚好在紧跟先前的代码转换的图象的第 N 个图象从译码器输入缓存器被读出以后的时刻。这样, 如果 N 是 1, 则将来的时刻就是刚好在当前被代码转换的图象从译码器输入缓存器被读出以后的时刻。然而, 如果 N 是 1, 则这相应于此此前描述的使用短期目标的方法。整数 N 最好相应于被包含在一组图象中的图象数(例如, 在 MPEG 中所规定的)。例如, N 可以是 12。

上述的将来时刻可以根据由与当前被代码转换的图象有关的“vbv 缓存器延时”规定的译码器输入缓存器延时 $\Delta Tdib$ 来进行计算。假定“现在”是刚好在当前要被代码转换的图象被写入到代码转换器输出缓存器之前的时刻。参照图 3, $PT(i)$ 是当前要被代码转换的图象, 则“现在”是时刻 Twt 。将来的时刻是在“现在”开始的时间范围的结束点, 该时间范

图具有的长度是译码器输入缓存器延时 ΔT_{dib} 和代码转换延时 ΔT_{trans} 加上 $N-1$ 个图象周期的和值。这样，将来的时刻是“现在”加上 ΔT_{dib} 加上 ΔT_{trans} 加上 $N-1$ 乘 ΔT_{pt} ， ΔT_{pt} 是图象周期。译码器输入缓存器延时 ΔT_{dib} 与代码转换延时 ΔT_{trans} 的和值标志 N 个图象的第一图象从译码器输入缓存器被读出的时刻。因此，由于图象是在每个图象周期从译码器输入缓存器被读出的，第 N 个图象就在第一个图象被读出以后的 $N-1$ 个图象周期被读出。

对于要被代码转换的 N 个接连的图象的一组目标是根据上述的时间范围、在这个时间范围期间对于代码转换的数据流的投影比特率、实际的代码转换器输出缓存器充满度、和在上述的将来时刻的想要的译码器输入缓存器充满度而被计算的。该组目标代表通过代码转换接连的 N 个图象应当理想地得到的数据量。也就是，如果该组目标被满足，则译码器输入缓存器充满度将是在将来时刻的想要的译码器输入缓存器充满度。

组目标被计算如下。在上述的时间范围期间将从代码转换器输出缓存器被代码转换到译码器输入缓存器的数据量是根据投影比特率被计算的。组目标是上述的数据量减去实际的代码转换器输出缓存器充满度以及减去想要的译码器输入缓存器充满度。例如，假定在所关心的时间范围中的比特率是常数并且等于 R 。那么，在该时间范围期间从代码转换器输出缓存器被代码转换到译码器输入缓存器的数据量将是 R 乘以时间范围的长度。这样，组目标等于 R 乘以由时间范围的长度减去实际的代码转换器输出缓存器充满度以及减去想要的译码器输入缓存器充满度。

对于要被代码转换的图象的图象目标是根据上述的组目标以以下的方式被计算的。首先计算最后 N 个代码转换的图象的复杂度的总和。图象目标是最后的具有与要被代码转换的图象相同的编码类型的被代码转换的图象的复杂度，它乘以组目标以及被除以复杂度的总和。这样，如果最近的代码转换的图象是相对较复杂的，意思是其复杂度具有高的数值，则要被代码转换的具有相同编码类型的接连的图象将具有相对较高的图象目标。

应当指出，以前几段给出的图象目标没有考虑图象复杂度，而是从短期内的实际的缓存器充满度和想要的缓存器充满度、即下一个图象直接得到的。这样，这个基于短期的图象目标是与在前面的段落中给出的基于长期的图象目标的严格比较。在前段中给出的基于长期的图象目标考虑到

某些图象可以比其它的更复杂,因此当以其它的图象为代价进行代码转换时,应当允许它们产生更多的比特。因此,可以防止量化参量 QP 受到相对较大的短期的起伏,这种起伏对于图象质量是不利的。

有利地,对基于长期的图象目标进行检验以使得确保: 如果通过代码转换图象而得到的数据量理想地等于这个图象目标,则译码器输入缓存器既不下溢也不上溢。图 4 和 5 分别显示了用于上溢和下溢的检验方法。如果图象目标会造成下溢,则把它减小以防止下溢。反之,如果图象目标会造成上溢,则把它增加以防止上溢。量化参量 QP 的基本值是通过把最后的代码转换的相同编码类型图象的复杂度除以已被检验的图象目标而得到的。

量化参量 QP 的基本值可被应用于要被代码转换的图象中的每个宏块。然而,根据图象质量逐个宏块地调整量化参量的数值一般是有利的。这可以以下面的方式来进行。MPEG 数据流包括控制码,它表示对于每个宏块曾被用来编码该宏块的量化参量的数值。这个数值(以后将被称为原始值)被用来得出相关系数。这样,对于每个宏块得出相关系数。要被应用来代码转换宏块的量化参量 QP 的值等于量化参量 QP 的基本值乘以宏块的相关系数。

相关系数可以以下面的方式被计算。每个原始值被除以取决于所考虑的宏块的编码类型的加权因子,以便得到加权的原始值。计算最后代码转换的图象的加权的原始值的平均值,以便得出平均的原始值。相关系数等于两倍的当前原始值加上平均原先值与当前的原始值加上两倍的平均原始值的比值。

此前和参照图 3-7 描述的确定量化参量 QP 的数值的各个不同方面共同地具有以下的特性。与要被代码转换的数据部分 PT 有关的译码器输入缓存器延时 ΔT_{dib} 被用来确定用于代码转换该数据部分 PT 的量化参量 QP。这样,在代码转换 MPEG 数据流时,与编码的图象有关的 vbv 缓存器延时被用来确定用于代码转换该图象的量化参量。这样做的理由,例如,可从图 3 得以了解。译码器输入缓存器延时 ΔT_{dib} 越大,则刚好在所关心的数据部分从译码器输入缓存器被读出之前译码器输入缓存器越被充满(F_{dib} 越大),反之亦然。由于译码器输入缓存器延时 ΔT_{dib} 被用来确定量化参量 QP,所以可以防止译码器输入缓存器超过上限(上溢),或低于下限(下溢)。而且,在针对当第 N 个接下来的数据部分从译码器输入缓存器被读

出时想要的译码器输入缓存器充满度的长期方法中,译码器输入缓存器延时 ΔT_{dib} 可被用来计算在编码 N 个接连的数据部分时得到的想要的代码转换的数据量。对于量化参量 QP 的适当的数值可从该想要的的数据量得到。

附图和前面对它们的描述都是为了显示本发明而不是限制本发明。

- 5 显然,有多种属于附属权利要求的范围内的替换方案。在这方面,将作出以下的有关说明。

在各个单元上,有多种物理上扩展功能的方式或功能性元件。在这方面,附图是非常概略的,每个附图只表示本发明的一个可能的实施例。因此,虽然附图显示了作为不同的方块的不同的功能性元件,但这决不排除

- 10 除某些功能性元件或所有功能性元件作为单个物理单元来实施。

在权利要求中,括号之间的任何参考符号将不认为是对所涉及的权利要求的限制。

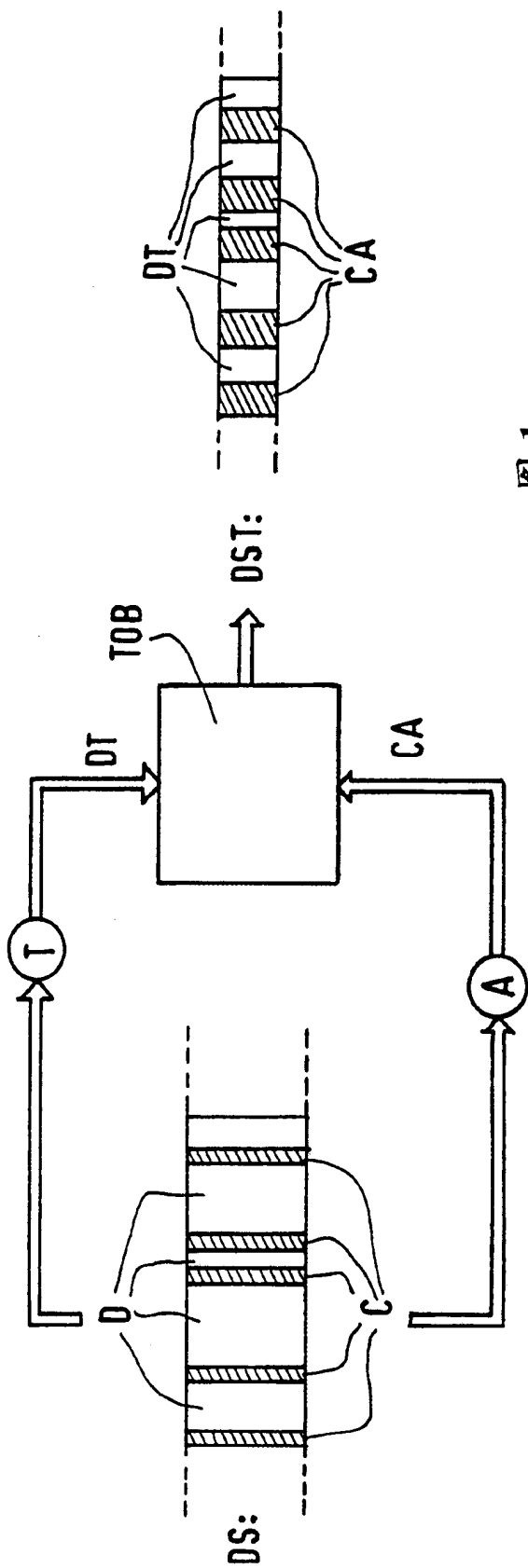


图 1

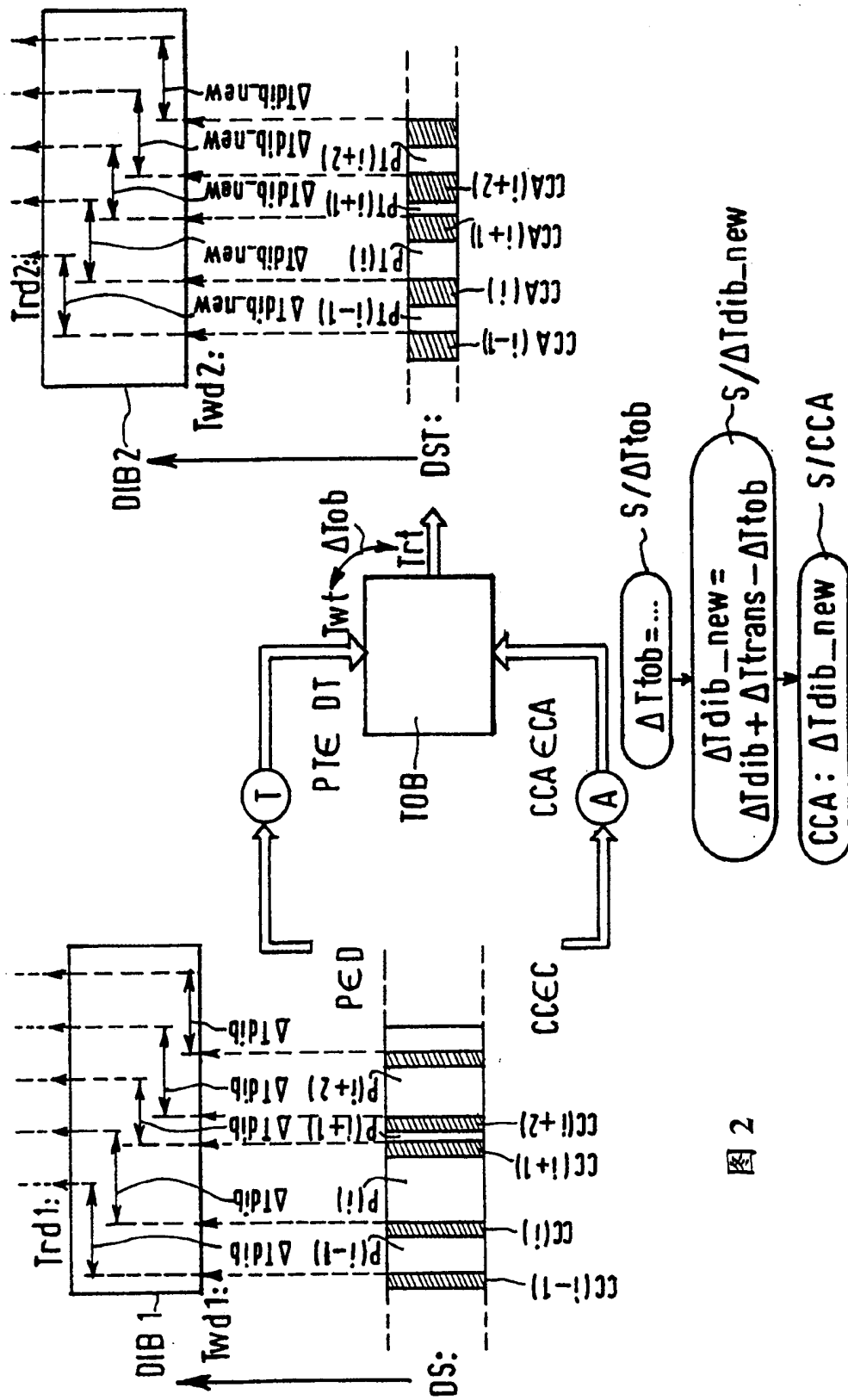


图 2

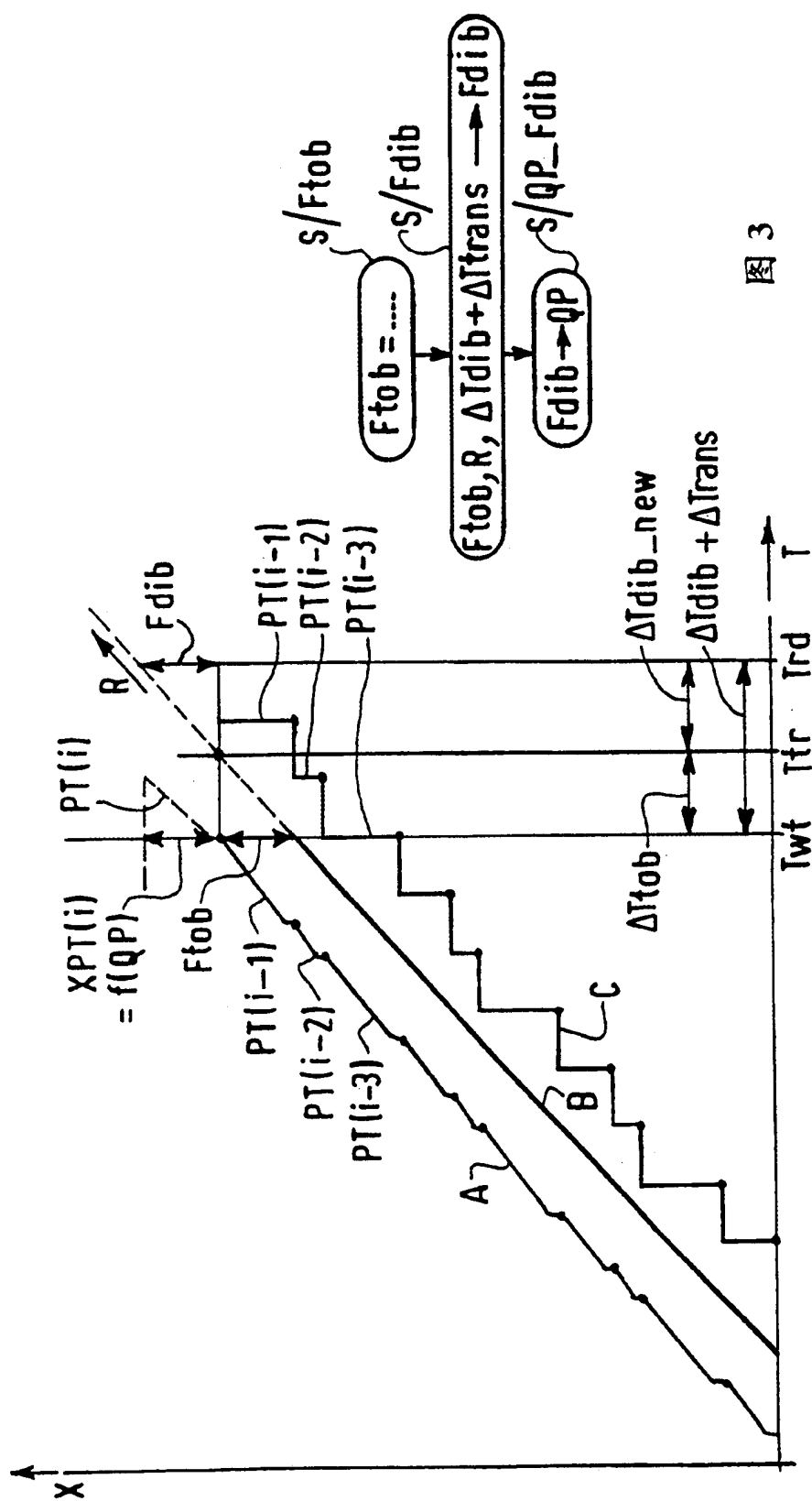


图 3

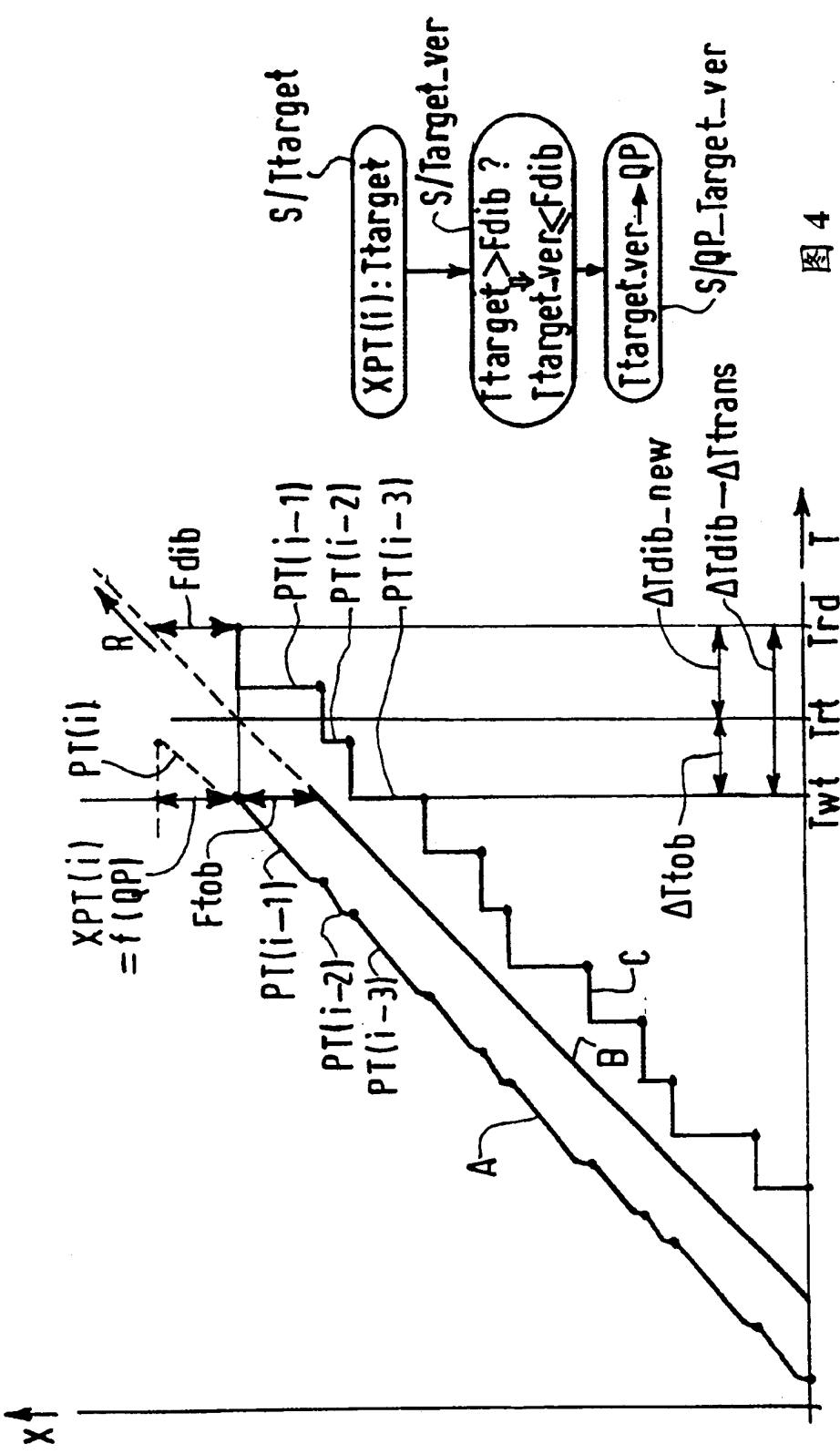


图 4

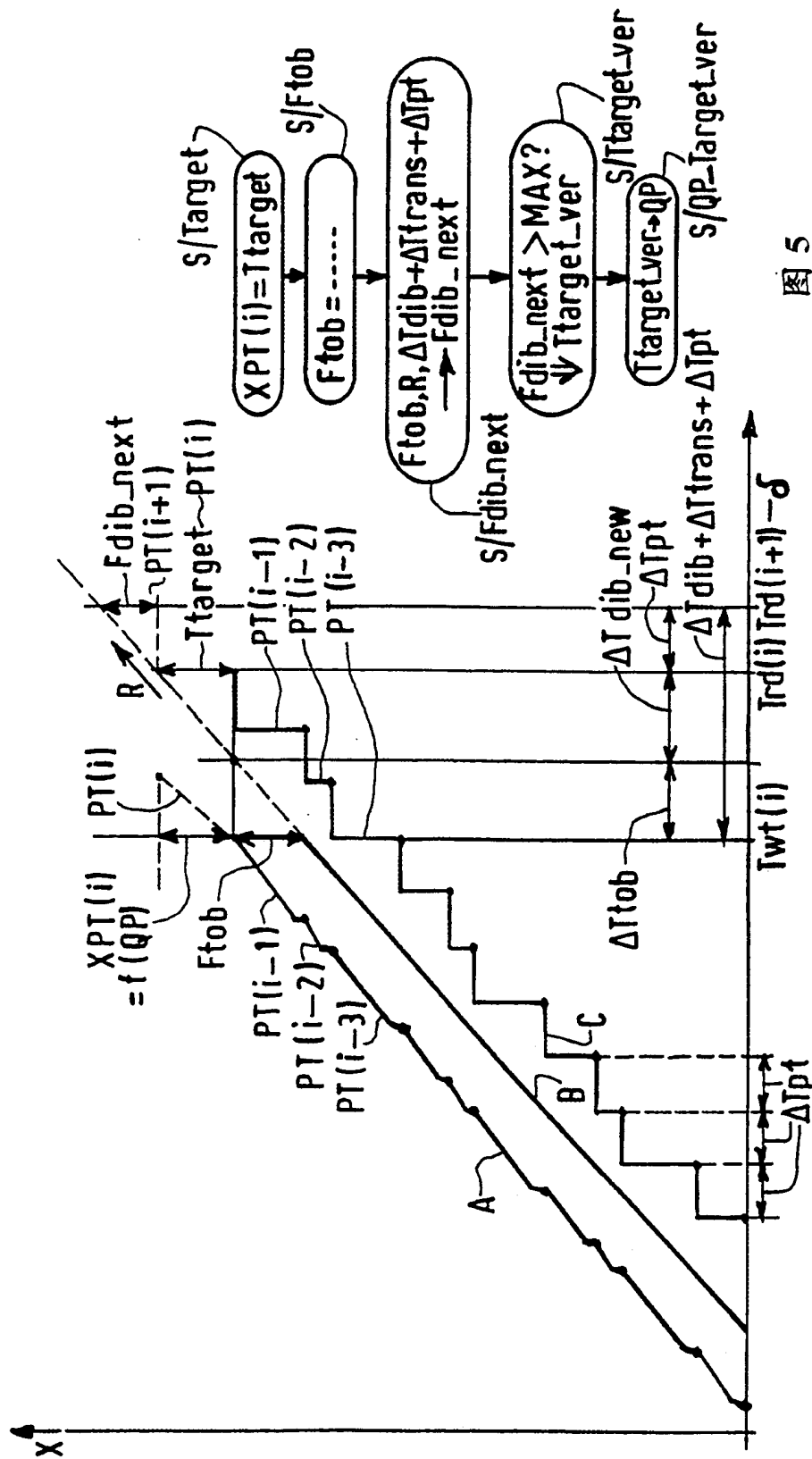


图 5

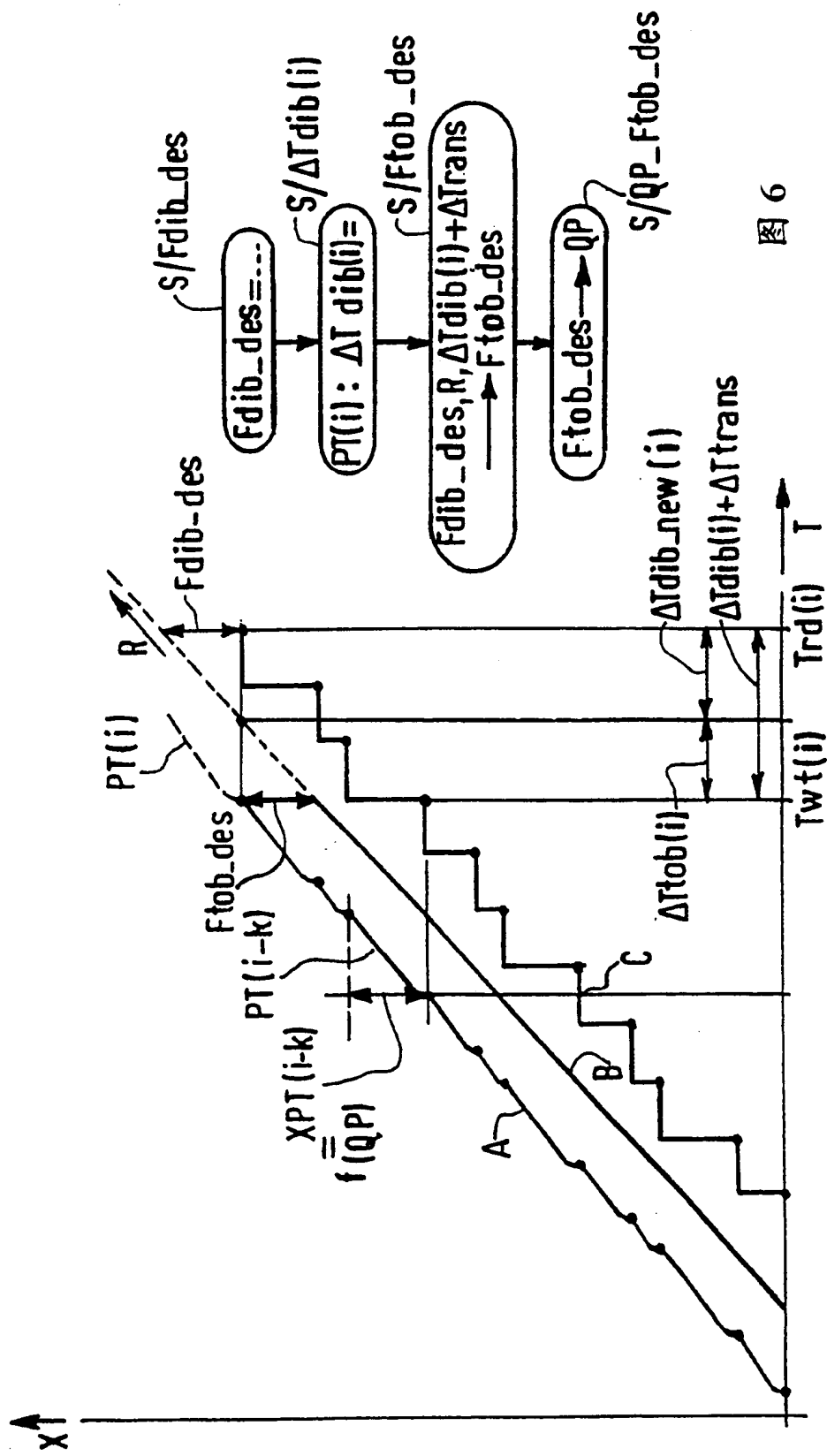


图 6

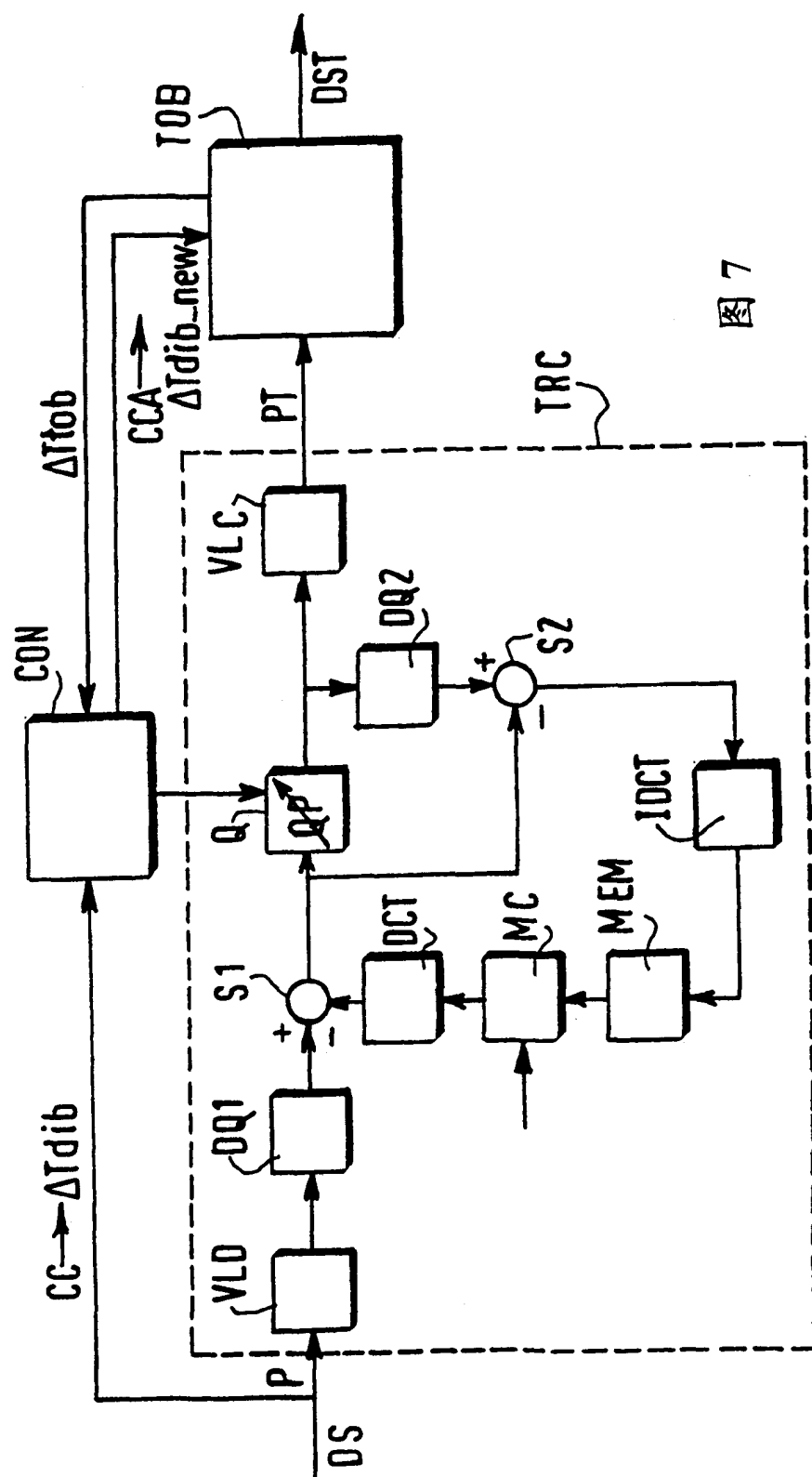


图 7