



## [12] 发明专利说明书

[21] ZL 专利号 99800869.9

[45] 授权公告日 2004 年 12 月 8 日

[11] 授权公告号 CN 1179574C

[22] 申请日 1999.2.11 [21] 申请号 99800869.9

[30] 优先权

[32] 1998.3.31 [33] EP [31] 98400759.1

[32] 1998.4.3 [33] EP [31] 98400802.9

[86] 国际申请 PCT/IB1999/000235 1999.2.11

[87] 国际公布 WO1999/051033 英 1999.10.7

[85] 进入国家阶段日期 2000.1.31

[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·拜罗伊尔

审查员 孙 慧

[74] 专利代理机构 中国专利代理(香港)有限公司

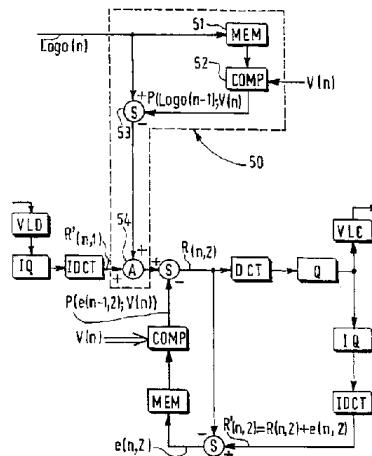
代理人 程天正 李亚非

权利要求书 1 页 说明书 9 页 附图 6 页

[54] 发明名称 在编码数据流中修改数据的方法和设备

## [57] 摘要

对编码数据流进行解码和重新编码的基本代码转换器可以用于在所说的流中做局部修改。 所说的修改的实现过程是：依据给定的修改数据流来规定预测数据流并在代码转换器的解码部分与编码部分之间将修改数据流和它的预测值之间的差值加到经过解码的流上去。 在改进的实现方案中，将对修改数据流做预处理，以使得用在解码后的流中得到所说的修改数据局部取代解码得到的流。



1. 将一个附加数据流插入一个编码的数据流的方法，该编码的数据流对应于被划分为子图象的连续图象，该方法包括以下步骤：

-对所述已编码的数据流进行解码，以便传送已解码的残差信号；

5 -对已解码的数据进行重新编码；

-伪预测，用于传送来自一个编码误差的一个预测信号和以前为每个相应的子图象确定的运动矢量，所述编码误差从重新编码步骤得到；

其特征在于该方法还包括步骤：

-根据附加数据和运动矢量计算运动补偿附加数据；

10 -将已解码的残差信号与附加数据和运动补偿附加数据之间的差值相加，伪预测步骤用于将所述相加步骤的结果和预定信号之间的差值传送到重新编码步骤。

2. 一种将一个编码的数据流进行代码转换的设备，该编码的数据流对应于被划分为子图象的连续图象，该设备包括：

15 -对所述已编码的数据流进行解码并传送已解码的残差信号的残差解码单元；

-对已解码的数据进行重新编码的编码与解码单元；

20 -伪预测单元，用于传送来自一个编码误差的一个预测信号和以前为每个相应的子图象确定的运动矢量，所述编码误差从编码与解码单元得到；

其特征在于，该设备还包括用于将一个附加数据流插入一个编码的数据流的装置，所述插入装置包括：

-用于根据附加数据和运动矢量计算运动补偿附加数据的计算装置；

25 -用于把所述附加数据减去所述运动补偿附加数据的结果加到已解码的残差信号的加法装置，伪预测单元用于将所述加法装置的结果和预测信号之间的差值传送到编码与解码单元。

## 在编码数据流中修改数据的方法和设备

### 技术领域

5 本发明涉及在经过编码的数据流中修改数据的方法，该编码的数据流与被分为子图象的连续图象相对应，该方法的步骤包括：

- 对所说的经过编码的数据流进行解码；
- 对解码后的数据流重新进行编码。

10 本发明还涉及用于实现所述方法的视频处理设备。本发明不仅可以在 MPEG-2 压缩领域中使用（MPEG-2 是 TV 广播的通用标准），比如当电视台希望在图象序列中加入他们自己的图标时就可以使用，更一般地，本发明可以应用在任何数字视频数据压缩系统中。

### 背景技术

15 在某些情况下，比如在进行传输之前可能需要将附加的数字数据加入到现有的经过编码的比特流中去，最简单的解决方案是先对所说的比特流进行解码，然后再进行所说的添加。接下来再对修改过的比特流重新进行编码和传输。不幸的是，通常情况下完全解码的代价是很昂贵的，因为它需要一个解码器和一个编码器。此外，用重新估计的模式和矢量重新进行编码时会损伤那些不需要加入图标的地方的图  
20 象质量。

### 发明内容

因此，本发明的第一目的是提出一种代价低廉的解决方案以便能在已有的经过编码的比特流中添加数据。

25 为达到这个目的，本发明涉及了一种在本说明书的开篇部分进行了描述的方法，该方法的特征在于它还包括下列步骤：

- 根据输入的修改数据流确定一个预测数据流；
- 在进行重新编码之前，将修改数据流与预测数据流之间的差值添加到经过解码的数据流上去。

30 为达到预期的效果，该方法还要借助混合步骤将要附加数据转换为一个特殊的校正信号，接下来使该信号与代码转换器的残留信号相加。

本发明的另一个目的是提出了一种对于所述数据添加具有更高质

量的改进的用以实现数据插入的方法。

为到达这个目的，本发明涉及了这样一种方法，其特征在于该方法还包括对修改数据流进行预处理的步骤，以便在解码后的数据流中可以用修改数据流来取代经过解码的数据流。

5 本发明的另一个目的是提出用于实现上述方法的设备。

为了达到这个目的，在第一种实现方案中，本发明涉及一种用于将数据添加入经过编码的视频数据流的设备，其特征在于它包括：

- (1) 代码转换器，它用于对所说的数据流进行解码和重新编码；
- (2) 相加分支，它用于确定：

10 (i) 预测信号，这要根据附加数据和在代码转换器中确定的运动矢量来实行；

(ii) 所说的附加数据与所说的预测信号之间的差值，以便在代码转换器的解码部分和重新编码部分之间将所述差值添加入所说的数据流。

15 在第二种实现方案中，本发明也涉及了一种用于将数据插入经过编码的视频数据流的设备，其特征在于它包括：

- (1) 代码转换器，它用于对所说的数据流进行解码和重新编码；
- (2) 插入分支，它用于确定：

(i) 预处理的信号，这要依据这些附加数据来实行；

20 (ii) 预测信号，这要依据经过预处理的信号和在代码转换器中规定的运动矢量来实行；

(iii) 所说的预处理的附加数据和所说的预测信号之间的差值，以便在代码转换器的解码部分和重新编码部分之间将所述差值插入所说的数据流。

25 在美国专利 5691986 中描述了一种用于将数据插入经过编码的数据流的方法和装置。在所述的情况下，所说的流事实上是基本数据流复接在一起形成的，所以必须对其进行解复接以便能从这些基本流中分离出一个流，并且还要对这个分离出来的流进行数据缩减操作，然后将所说的数据插入到缩减后的数据流中去。本发明有一个明确的目的，这就是在图象中添加数据，或者是象在一个改进的实施方案中那样，用另一图象的数据取代图象的一个局部部分。

#### 附图说明

本发明的这些方面和其它方面将参考下文中将要描述的实施方式来进行阐述和说明。

在附图中：

图 1 和图 2 给出了常规的视频解码器和编码器；

5 图 3 给出了用于在输入比特流中添加入一个图标链式传送过程的一种可能方案；

图 4 画出了一个没有图标添加器的公知转换器的略图；

图 5 所示的是一个根据本发明的设有图标添加器的代码转换器；

图 6 所示的也是一个根据本发明设有图标插入设备的代码转换器；

10 图 7 显示了在所说的图标插入设备中所进行的计算；

图 8 到图 10 给出了对亮度和色度分量所做的处理，并且

图 11 和图 12 给出了在完成了所说的处理之后的图标剪辑；

具体实施方式

一个传统的如图 1 所示的视频解码器包括解码通道 12 和运动补偿通道 14。解码通道 12 包括串联在一起的变长解码电路 1、反量化电路 2 和反频率变换电路 3 (分别是：VLD, IQ, IDCT)。运动补偿电路通道 14 包括接收解码器输出信号的图象存储器 4、运动补偿电路 5 (所说的补偿要考虑存储器 4 的输出信号和解码器接收到的运动矢量 V (n))、以及用于电路 3 和电路 5 的输出信号的加法器 6 (分别是 MEM, COMP, A)。通过将预测值 (电路 5 的输出) 与在解码通道 12 的电路 3 的输出中得到的经过解码的残差信号相加可以重建解码器的输出图象 (该图象也要送给图象存储器 4)。

一个如图 2 所示的常规视频编码器包括编码与解码通道 13 和预测通道 11。编码与解码通道 13 包括离散余弦变换电路 25、量化电路 26、变长编码电路 27 (分别为：DCT, Q, VLC)，以及在电路 26 的输出端串联起来的反量化电路 28 和反离散余弦变换电路 29 (分别是：IQ 和 IDCT)。预测通道 11 用于从输入信号中减去经过运动补偿的预测信号，它包括用于在预测之前重建图象的加法器 21、图象存储器 22、运动补偿电路 23 和减法器 24 (分别是：A, MEM, COMP, S)。所说的补偿要考虑前面经过估计的运动矢量 V (n)。

图 3 给出了一种可能的方法和设备，它能在链式传送过程中的输入比特流中加入象图标这样的数据。所说的链包括第一编码器 31 (称

为“编码器 1”）、子系统 305、和处在传输过程之后的解码器 35（“解码器 2”）。该子系统 305 用于在编码器 31 的输出端上得到的经过编码的比特流中加入图标。位于所说的编码器和解码器之间的子系统本身包括如图所示的解码器 32（“解码器 1”）、图标加法器 33 和编码器 34（“编码器 2”）。于是，从所说的解码器 32 和所说的编码器 34 开始，利用它们的互补性可以做一些简化，以便实现根据本发明的代码转换器的基本功能。

根据图 2，对第一编码器 31 来说实际上有：

$$R(n, 1) = I(n) - P(I'(n-1), 1; V(n)) \quad (1)$$

其中在  $R(\cdot)$  和  $P(\cdot)$  中的下标 (, 1) 表示“第一”编码器 31 (= 编码器 1)， $I(n)$  是编码器的原始视频输入， $P(I'(n-1), 1; V(n))$  是预测信号，它通过将运动矢量  $V(n)$  加到先前“解码”后的图象  $I'(n-1)$  上来计算而得到，从原始的输入比特流  $I(n)$  中减去  $P(I'(n-1), 1; V(n))$  的目的是为了得到要被进行编码的残差信号  $R(n)$ ，而  $R(n)$  表示残差信号。需要注意的是在编码器的预测通道输入端的信号  $R'(n)$  与  $R(n)$  相差一个被称为编码误差的值  $e(n)$ ，由于  $R'(n)$  等于  $R(n) + e(n)$ ，所以在预测通道中加法器的输出端上的信号应该是  $I'(n) = I(n) + e(n)$ 。

对于跟在编码器 31 之后的第一解码器 32 而言，根据图 1，它有类似的表示：

$$I'(n, 1) = R'(n, 1) + P(I'(n-1), 1; V(n)) \quad (2)$$

其中在  $I'(\cdot)$ ， $R'(\cdot)$  和  $P(\cdot)$  中的下标 (, 1) 表示“第一”解码器 32， $R'(n)$  是解码后的残差信号， $P(I'(n-1), 1; V(n))$  是一个预测信号，它将被用来与  $R'(n)$  相加，而  $I'(n)$  表示解码器的输出。正如上面所示， $I'(n, 1)$  也可以写成这种形式：

$$I'(n, 1) = I(n) + e(n, 1)$$

这就是说，解码器的输出是原始输入信号  $I(n)$  与在对  $I(n)$  进行编码操作期间产生的编码误差  $e(n)$  之和。所以，在图标加法器 33 的输出端有：

$$J'(n, 1) = I'(n, 1) + Logo(n) \quad (3)$$

其中  $I'(n)$  是解码器 32 的输出，而  $Logo(n)$  是将要与主比特流相加的数据（比如说一个图标）。输出的结果  $J'(n)$  将要被送给编码器

34.

对第二个编码器 34 而言, 有下式 (与前面第一个编码器 31 的情形类似):

$$R(n, 2) = J'(n, 1) - P(J'(n-1), 2; V(n)) \quad (4)$$

其中在  $R(\cdot)$  和  $P(\cdot)$  中的下标 (2) 现在表示 "第二" 编码器 34,  $P(J'(n-1), 2; V(n))$  是一个预测值, 将要从图标加法器 33 的输出  $J'(n-1)$  中减去这个值, 以便获得要被编码的残差信号, 而  $R(n)$  表示所说的残差信号。

最后, 对第二个解码器 35 而言, 有 (与前面第一个解码器 32 的情形类似):

$$J'(n, 2) = R'(n, 2) + P(J'(n-1), 2; V(n)) \quad (5)$$

其中在  $J'(\cdot)$ 、 $R'(\cdot)$  和  $P(\cdot)$  中的下标 (2) 现在表示 "第二" 解码器 35,  $R'(n)$  是解码后的残差信号,  $P(J'(n-1), 2; V(n))$  是要用来与  $R'(n)$  相加的预测值, 而  $J'(n)$  表示解码器的输出。就解码器 32 而言,  $J'(n, 2)$  可以写成下面的形式:

$$\begin{aligned} J'(n, 2) &= J'(n, 1) + e(n, 2) \\ J'(n, 2) &= I'(n, 1) + \text{Logo}(n) + e(n, 2) \\ J'(n, 2) &= I(n) + e(n, 1) + \text{Logo}(n) + e(n, 2) \end{aligned} \quad (6)$$

这说明传送链的输出信号  $J'(n, 2)$  实际上就等于原始输入信号  $I(n)$ 、第一编码误差 (在第一编码器和解码器中编码/解码)、第二编码误差 (在第二编码器/解码器中编码/解码)、与附加数据之和。

于是, 利用运动补偿算子的线性特性, 它可以写为:

$$P(J'(n-1), 2; V(n)) = P[((J'(n-1), 1) + (e(n-1), 2)); V(n)] \quad (7)$$

于是可以将 (4) 的关系式写为这种形式:

$$R(n, 2) = J'(n, 1) - P[((J'(n-1), 1) + (e(n-1), 2)); V(n)] \quad (8)$$

再次利用补偿算子的线性特性, 得到:

$$P(J'(n-1), 1; V(n)) = P((I'(n-1) + \text{Logo}(n-1)), 1; V(n))$$

或:

$$P(J'(n-1), 1; V(n)) = P(I'(n-1), 1; V(n)) + P(\text{Logo}(n-1), 1; V(n)) \quad (9)$$

于是关系式 (8) 变为:

$$\begin{aligned} R(n, 2) &= I'(n, 1) + \text{Logo}(n) - P(e(n-1), 2; V(n)) \\ &\quad - P(I'(n-1), 1; V(n)) - P(\text{Logo}(n-1), 1; V(n)) \end{aligned} \quad (10)$$

或是由关系式 (2) :

$$\begin{aligned} R(n, 2) = & R'(n-1) - P(e(n-1), 2; V(n)) \\ & + \text{Logo}(n) - P(e(n-1), 2; V(n)) \\ & - P(I'(n-1), 1; V(n)) - P(\text{Logo}(n-1), 1; V(n)) \end{aligned} \quad (11)$$

5 于是最终得到下面的等式 (12) :

$$\begin{aligned} R(n, 2) = & R'(n-1) - P(e(n-1), 2; V(n)) \\ & + \text{Logo}(n) - P(\text{Logo}(n-1); V(n)) \end{aligned} \quad (12)$$

这就是根据本发明提出的具有图标加法器的代码转换器最终的系统等式。

10 没有图标加法器的代码转换器的一般概貌首先在图 4 中提到 (为了与图 5 进行比较). 它包括残差解码分支 41 (变长解码 VLD+反量化 IQ + 反离散余弦变换 IDCT), 编码与解码分支 42 (离散余弦变换 DCT+量化 Q + 变长编码 VLC ; 反量化 IQ + 反离散余弦变换 IDCT), 以及被称为伪预测分支的中间分支 43 (第一个减法器 S+存储器 MEM+基于运动矢量 V (n) 的运动补偿 COMP+第二个减法器 S ). 之所以要这样称分支 43 是因为它并不是如同在基本编码器中那种严格的经典预测, 在这里加法器被一个减法器取代. 图 4 给出了在前面提到的信号 R' (n, 1), R (n, 2), R' (n, 2), e (n, 2), V (n), P (e (n-1), 2; V (n)).

20 接下来在图 5 中给出了与本发明相应的代码转换器的方案, 比如具有图标添加器的代码转换器. 其中, 与图 4 相比, 相同的部分均用相同的方式来标识. 附加部分是一个图标添加分支 50, 它包括用于接收要添加的图标 (信号 Logo (n)) 的存储器 MEM51; 运动补偿电路 COMP52, 它接收存储器 51 的输出和矢量 V (n) 并发送预测数据流; 减法器 S53, 它提供原始信号 Logo (n) 与在电路 52 输出端得到的经过运动补偿的值 P (Logo (n-1); V (n)) (预测后的数据流) 之间的差值; 以及加法器 54, 它用于将所说的减法器 53 的输出信号送入主比特流 (对应于连续图象的完整序列). 因此, 图标的添加是通过向输入比特流中加入残差来实现的, 而这个残差是通过将图标减去经过运动补偿的图标预测值之后得到的, 该运动补偿的图标的预测值要根据包含有以前保存的图标的参考图象, 并且要使用与主比特流相同的模式和矢量.

30 上面提到的方法和装置还可以进一步改进. 事实上关注的焦点不

是添加、而是在不需要修改图象其它部分的条件下将附加数据（比如图标）插入主视频比特流中去。

数据的这种插入是一种非线性处理过程，也就是用所说的数据替代一副图象（或图象序列）的某个区域，要求完全访问原始象素（=图象元素）。所以对高质量的图标插入来说，需要对输入比特流进行完全解码。根据本发明，这样的完全解码将被一种低成本的粗略的解码所替代（这里是对 MPEG-2 标准的情形进行描述）。

图 6 给出了依据本发明的相对应的代码转换器（即具有图标插入器的代码转换器）的方案，其中，与图 4 相同的部分采用了相同的表示方式。附加的部分（与图 4 相比）是一个图标插入设备 400，现在将对它做详细的描述。

在这个图标插入设备中实现了四个步骤。第一个是在估计步骤 410 中完成的称为 DC 估计的步骤。MPEG 序列的每一幅图象被细分为称之为宏块的运动补偿单元。此外，在 MPEG 标准中，有三种图象类型：I（或帧内）图象，它的编码不需要参考其它图象；P（或预测）图象，它的编码需要参考过去的图象（I 或 P），以及 B 图象（或双向预测），该图象的编码却需要参考过去的图象和将来的图象（I 或 P）。这些 I 和 P 图象将被称为参考帧。在步骤 410 中，每个宏块的平均值将根据参考帧（过去保存在相关的缓冲存储器 401 和 402 中）的平均值和当前输入残差（图 4 中的信号  $R'(n, 1)$ ）的当前平均值进行估计。如果  $DCP(b, Y)$  表示在分量  $Y$  中宏块  $b$  的预测值  $PR(b, Y)$  的平均值（图 7 中画出了在隔行宏块情况下的这种计算方法：图中画出了在两个参考图象中的每一图象中的宏块、它的预测值  $PR(b, Y)$  以及它的平均值  $DCP(b, Y)$ ），如果  $ACR(b, Y)$  表示同一宏块在同一分量  $Y$  中的残差（即代码转换器的残差解码分支 41 的输出端上被称为  $R'(n, 1)$  的信号）的平均值，并且如果  $MDC(b, Y)$  表示同一宏块  $b$  在同一分量  $Y$  中的估计平均值，于是有：

$$MDC(b, Y) = DCP(b, Y) + ACR(b, Y) \quad (13)$$

虽然所给出的描述是针对分量  $Y$  的，事实上只要将色度分量  $U, V$  中的任何一个取代  $Y$  的位置就可以得到相同的关系式。

可以看到：只要假定在代码转换器输入端的输入比特流具有标准的广播 GOP 持续长度（比如  $N=12$ ）、以使得预测图象能够用帧内图象进

行充分的刷新，那么这种预测就不会带来明显的漂移（在此，一个 GOP 或图象组是由一幅 I 图象和它后面另一幅 I 图象之间的连续图象组成的，后面的 I 图象本身并不属于前面的 GOP，而 N 被定义为 GOP 的大小，在 MPEG 标准中 N 最常用的值是 12）。

5 第二个步骤是图标处理步骤，它在处理步骤 420 中完成。对图标象素值进行的处理实质上是为了保证图标的可视性并与图象内容保持一致。实际上是希望：

$$MV(CP) + MV(Logo(CP)) = RQM(CP) \quad (14)$$

其中 CP 表示亮度分量 Y 或色度分量 (U, V) 中的任何一种；MV(CP) 表示在 CP 分量中图标要插入的原始区域的平均值（这些值要根据上面提到的关系式 13 来估计）；MV(Logo(CP)) 是被处理过的图标的平均值，而 RQM(CP) 对应于图标插入后在图标区域所需要的图象的平均值。于是，经过图标处理在宏块 b 的位置上要插入的图标的象素值 Logo(b, CP) 可以用关系式 (15) 来表示：

15 
$$\begin{aligned} Logo(b, CP) &= Logo\_ori(b, CP) + RQM(CP) \\ &\quad - MV(CP) - MV\_Logo\_ori(CP) \end{aligned} \quad (15)$$

其中 Logo\_ori(b, CP) 表示在宏块 b 的位置上要插入的图标的原始象素值，而 MV\_Logo\_ori(CP) 表示 CP 分量原始图标的平均值。

图 8 到图 10 给出了图标的处理过程：图 8 给出了插入之前的原始图标信号 Logo\_ori(b, CP)、它的平均值 MV\_Logo\_ori(CP)、RQM(CP) 和 MV(CP)，而图 9 给出了所需要的  $(MV(CP) + Logo(b, CP)) = RQM(CP)$ ，图 10 画出了为了达到所需要的结果而进行调整的原理（正如图中所示，图标信号值有一个平移，平移的值等于  $(MV(CP) - RQM(CP)) + MV\_Logo\_ori(CP)$ ）。

25 对色度分量来说，需要让图标与原始图标保持一致，也就是说在将经过处理的图标插入图象时必须要保持原始图标的颜色。对色度分量 U 和 V 而言，RQM(CP) 就是 MV\_Logo\_ori(CP)。对亮度而言，可以选择与原始图象更好的一致性（即  $RQM(Y) = MV\_Logo\_ori(Y)$ ）或是选择可视性（这样，图标在亮的区域显得暗，或者反之亦然）。就后一种情况来说， $RQM(Y) = 128$ 。

第三个步骤是图标的裁剪步骤，这是在裁剪步骤 430 中完成的。根据 MPEG-2 标准，在解码器端，保存在存储器中的重建图象将被在 0

到 255 之间进行裁剪。在编码器端由于没有对输入的比特流进行完全解码，所以不可能在编码器端进行同样的操作。于是裁剪步骤将以宏块为单位来进行。虽然不知道确切的像素值，但由于在裁剪时采用了边界 MG 以使得这种情况得到的补偿。这样，对每个分量 Y、U 或 V（通常被称为 CP），Logo (b, CP) 都要被裁剪，于是有：

$$0 + MG < MV(b, CP) + Logo(b, CP) < 255 - MG \quad (16)$$

由此可得：

$$MG - MV(b, CP) < Logo(b, CP) \quad (17)$$

$$Logo(b, CP) < 255 - MG - MV(b, CP) \quad (18)$$

图 11 和图 12 给出了图标裁剪的方法，其中图 11 表示了 MV(b, CP)，  
MV(b, CP) + Logo(b, CP)，以及裁剪界限 MG 和 255 - MG，因此它  
表示了关系式 (16)，而图 12 类似地表示了关系式 (17) 和 (18)。

第四个步骤是图标添加步骤，它在图标添加步骤 440 中完成，这与图 5 中的图标添加分支 50 相同。所说的步骤 440 包括存储器，它用于接收在裁剪步骤 430 输出端上得到的裁剪后的图标；运动补偿电路，它用于接收所说的存储器的输出以及在代码转换器中确定的运动矢量 V (n)；减法器（用于获得经过裁剪的图标与经过运动补偿的图标之间的差值，前者可在图标插入设备 400 的处理部分的输出端得到，而后者可在运动补偿电路的输出端得到）、和加法器（用于将所说的减法器的输出信号插入主比特流）。这四个部件（存储器、运动补偿电路、减法器、加法器）与图 5 中 51 到 54 这四个部件相似。因此图标插入是通过将残差加入到输入比特流中去的方式来实现的，而该残差要通过确定图标（经所述的预处理的图标）与运动补偿后的预测值之间的差值来得到，其中该运动补偿后的预测值要通过所说的经过处理的图标并采用与主输入比特流相同的运动矢量来获得。

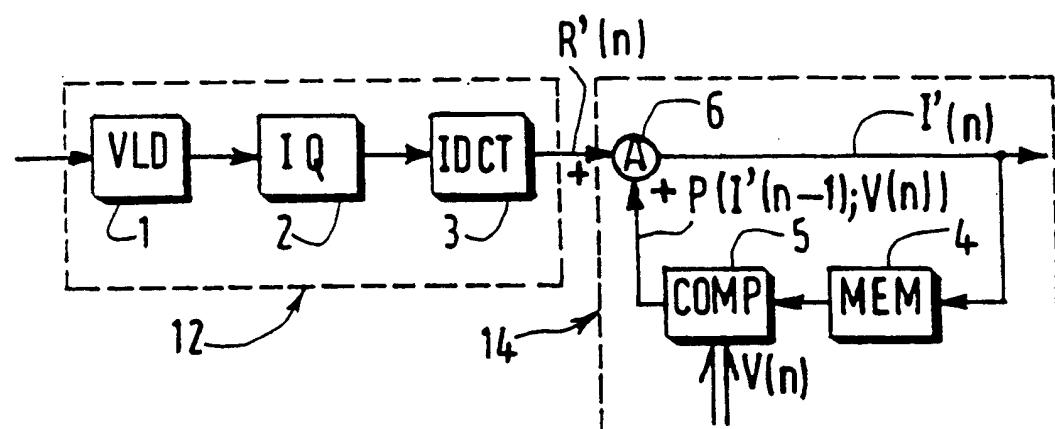


图 1

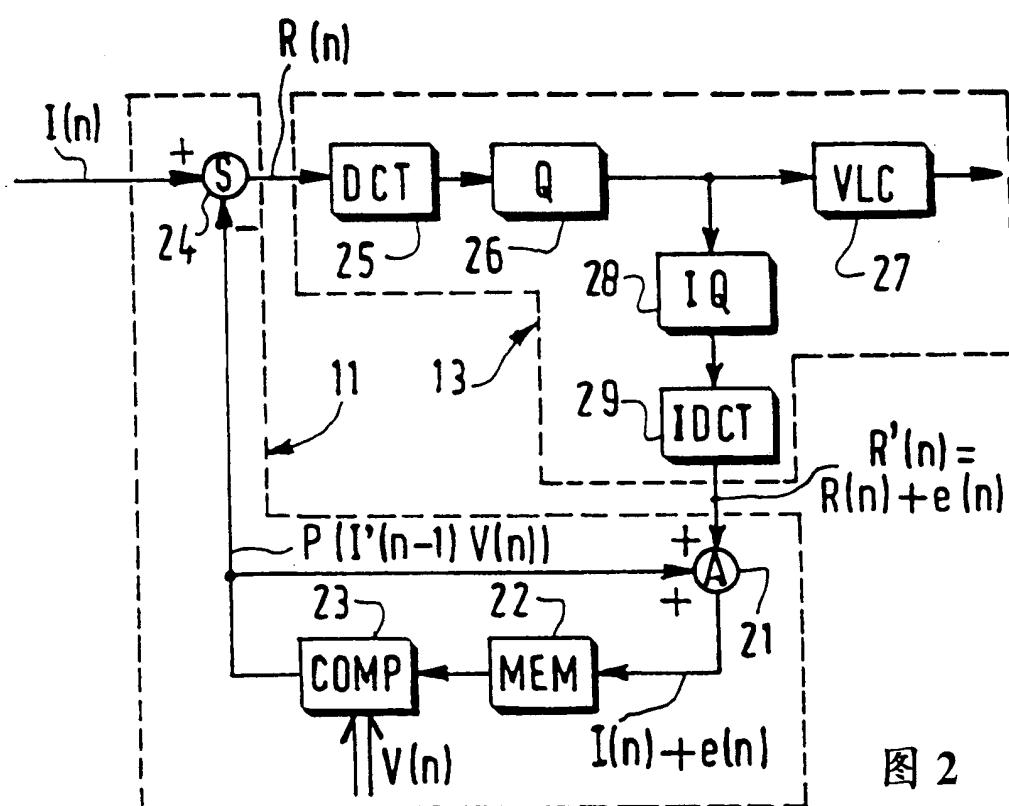


图 2

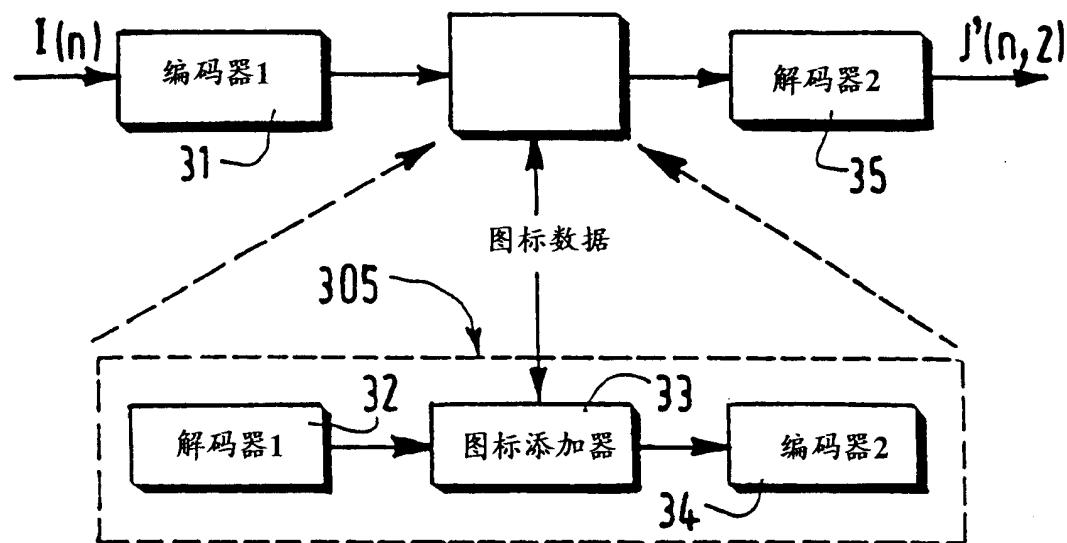


图 3

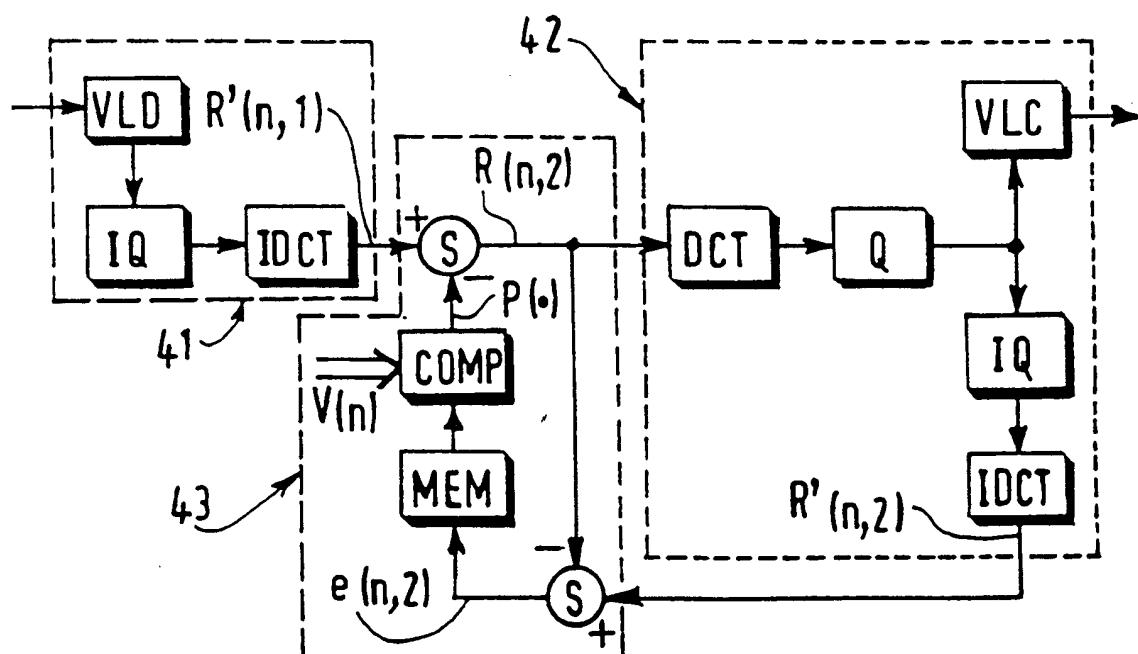


图 4

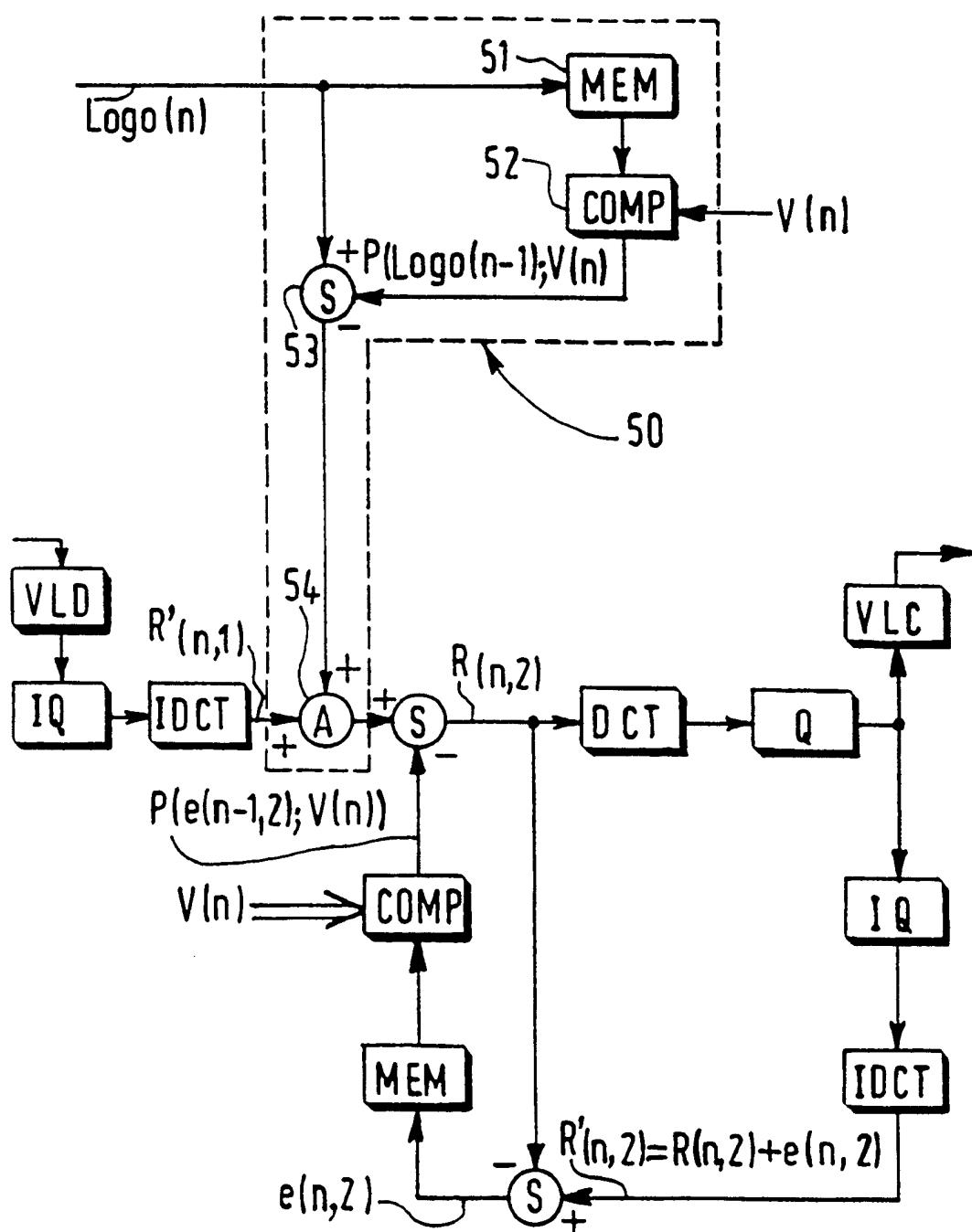


图 5

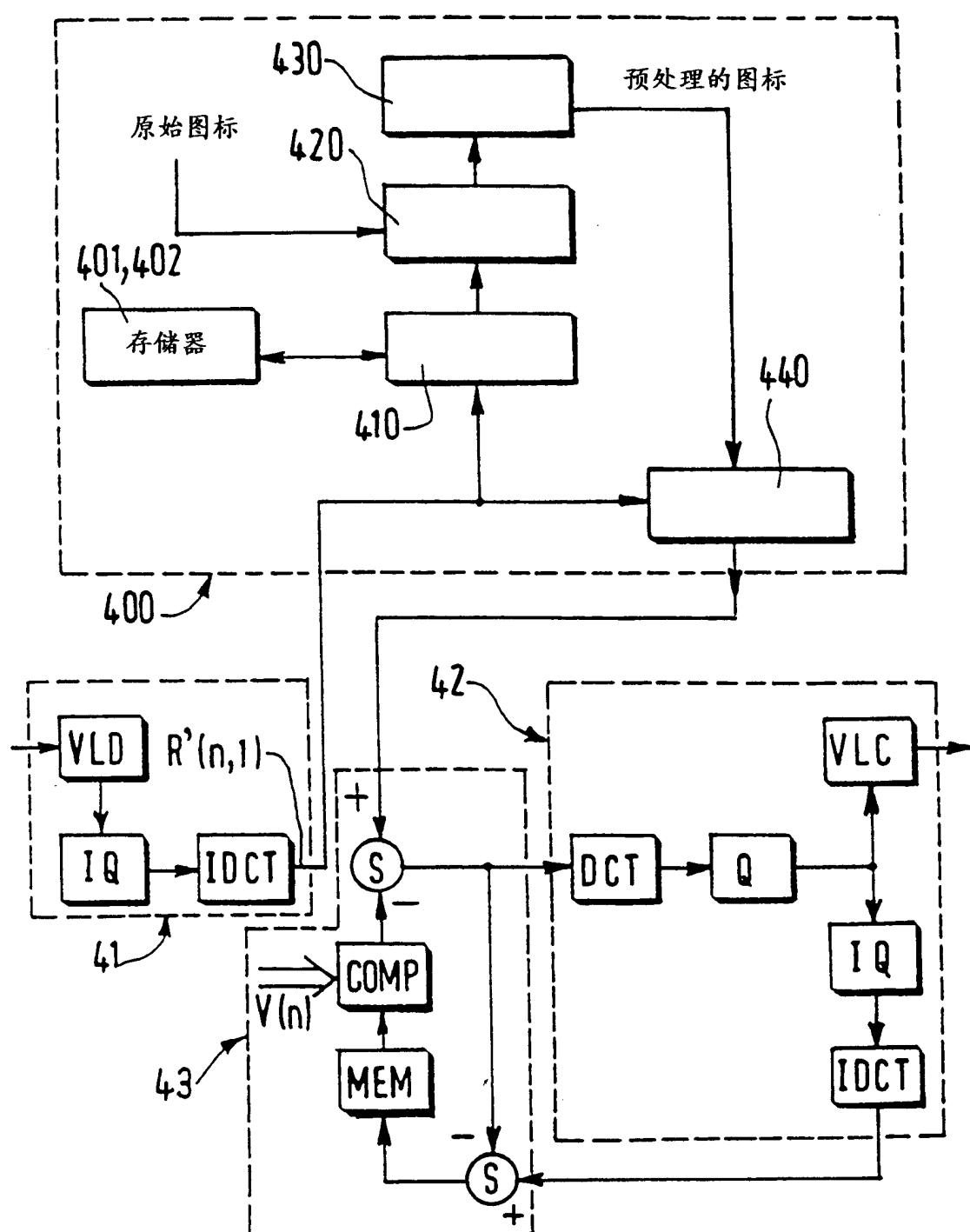


图 6

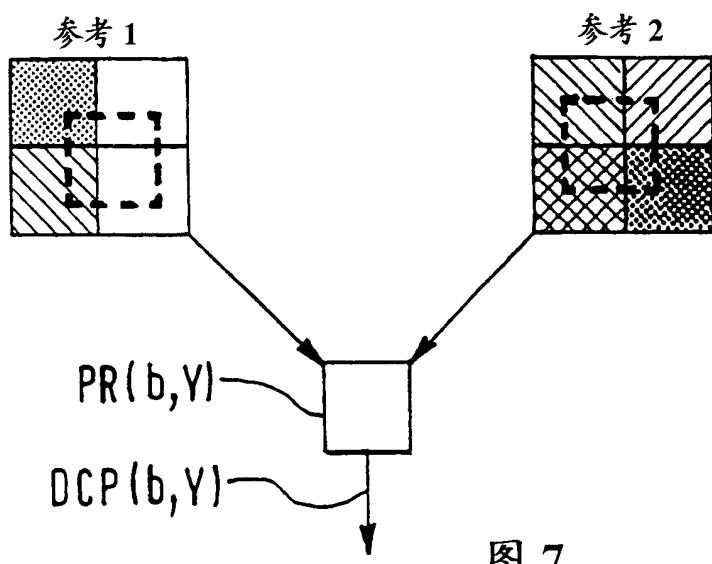


图 7

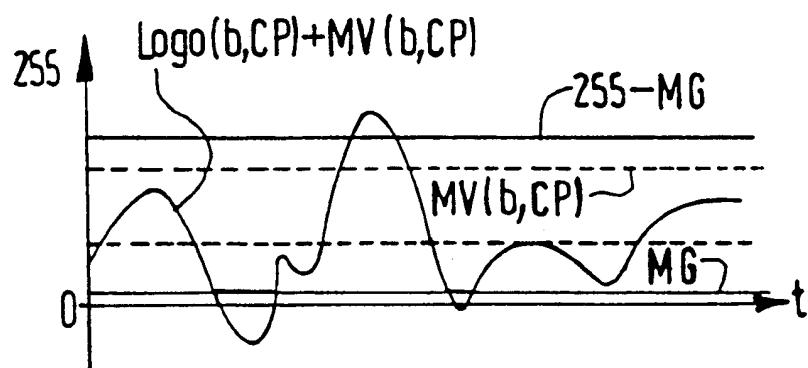


图 11

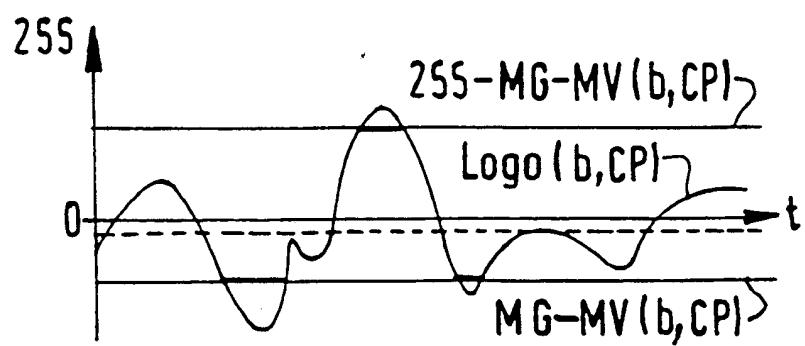


图 12

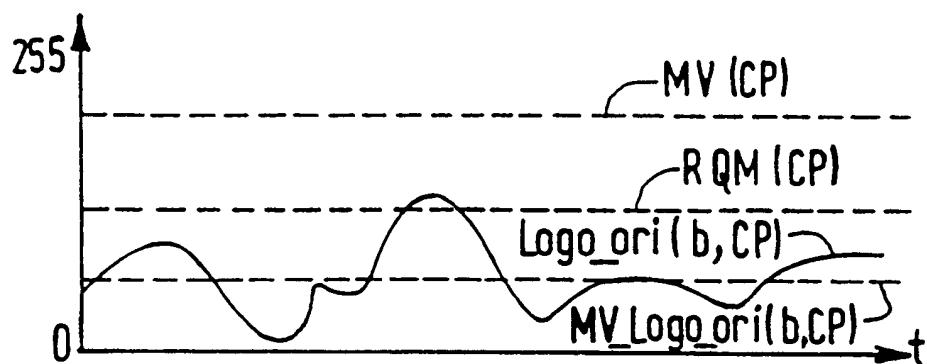


图 8

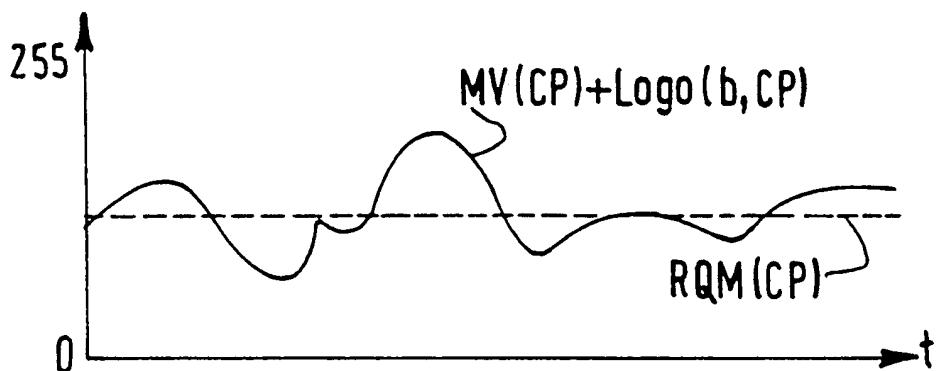


图 9

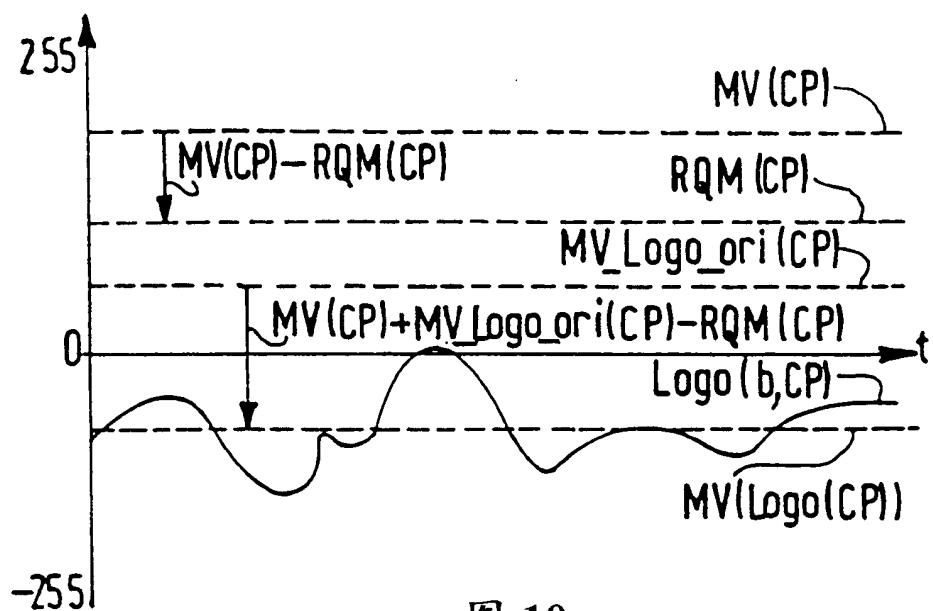


图 10