

权 利 要 求 书

1、一种图形信号编码方法，其中，通过把非间行图形转换成多组其中的每一组都开始于每个间行图形第一场的场图形所产生的图形串和通过把非间行图形转换成多组其中的每一组都开始于每个间行图形第二场的场图形所产生的另一图形串被相互结合成一个用于编码的输入图形信号，所述方法包括：

检测步骤，用于在所述的输入图形信号中检测不构成非间行图形的单个帧；和

编码步骤，用于将所述输入图形信号的图形串重新构成非间行图形串，并对所产生的非间行图形串进行编码。

2、如权利要求1的图形信号编码方法，其中，在所述的检测步骤中，通过检查输入图形信号的场和相邻场之间的相关性来检测不构成非间行图形的单个场。

3、如权利要求1的图形信号编码方法，其中，在所述的检测步骤中，根据第一场第 i 行和第二场 j 行之间的相关性以及第二场第 j 行和第一场第 $j+1$ 行之间的相关性来获得配对相邻场之间的相关性，并根据该相关性来检测所述单个场。

4、如权利要求1的图形信号编码方法，其中，在所述检测步骤中，当前场和多个过去或未来场之间的相关性被用于检查当前场是否是一个单个场。

5、如权利要求1的图形信号编码方法，其中，在所述的检测步骤中，删除该检测到的单个场，产生用于规定该单个场已经被删除

的第一标记，并产生第二标记，用于规定对于每个图形编码帧而言，在图形中首先显示第一场或第二场中的哪一个场。

6、如权利要求5的图形信号编码方法，其中，在所述的检测步骤中，所述的第一标记被附加到与被删除场暂时相邻并和被删除的场具有较强的相关性的过去或未来帧的标题信息上。

7、如权利要求5的图形信号编码方法，其中，在所述的检测步骤中，在目标图形被双向预测编码的情况下，禁止将用于规定单个场已被删除的第一标记设定成1，并将该第一标记作为标题信息附加到正在被编码的帧上。

8、如权利要求6的图形信号编码方法，其中，在所述的编码步骤中，所述第一和第二标记被编码成用于编码的每一帧的标题信息。

9、如权利要求8的图形信号编码方法，其中，若在所述的编码步骤中，用于规定单个场已被删除的第一标记被设置成1，并作为标题信息附加到正在被编码的一个帧上，所讨论的图形被帧内编码或正向预测编码。

10、一种图形信号编码方法，其中，通过把非间行图形转换成多组 其中的每一组都开始于每一间行图形第一场的场图形所产生的一个图形串和通过把非间行图形转换成多组 其中的每一组都开始于每一间行图形第二场的场图形所产生的的另一图形串彼此相互结合成为一个用于编码的输入图形信号；该方法包括：

检测步骤，用于根据所述的输入图形信号检测从第一场或第二场中的哪一个场开始了与非间行图形之一相对应的场图形组；和

编码步骤，用于把所述输入图形信号的图形串再构成为非间行图形串并对所产生的非间行图形串进行编码。

11、如权利要求10的图形信号编码方法，其中，在所述的检测步骤中，开始于第一场的两个配对场和开始于第二场的两个配对场之间的相关性被彼此加以比较，在每两个配对场中都包含有用作基准场的输入图形信号的预置场，具有较高相关性的两个配对场被判定为与规则非间行图形相对应的配对场，并输出判定结果。

12、如权利要求11的图形信号编码方法，其中，在所述检测步骤中，构成两个场的一个帧的相邻行之间差值绝对值的和被用作相关性，并且，具有较小和值的配对场被判定为与规则非间行图形相对应的配对场。

13、如权利要求10的图形信号编码方法，其中，在所述的再构成步骤中，输出用于规定第一场或第二场中的哪一场是与非间行图形相对应的输入图形信号图形串的一组配对场的标记。

14、一种用于对具有构成非间行图形的第一和第二场不规则结合的输入图形信号进行编码的图形信号编码方法，包括：

检查步骤，用于检查在输入图形信号中一个给定场和相邻场之间的相关性；

存贮步骤，用于存贮检测的结果；和

判定步骤，用于根据所存贮的检测结果来判断构成输入图形信号图形串非间行图形的场的结合。

15、如权利要求14的图形信号编码方法，其中，在所述的检测步骤中，根据第一场第 i 行和第二场第 i 行之间的相关性以及第二场第 j 行和第一场第 $j+1$ 行之间的相关性获得配对相邻场之间的相关性，并根据相关性检测所述单个场。

16、如权利要求14的图形信号编码方法，其中，通过把原始的

24图形-帧-图形转换成每秒25帧视频信号来获得所述的输入图形信号。

17、一种用于对具有构成非间行图形的第一场和第二场不规则结合的输入图形信号进行编码的图形信号编码方法，包括：

从输入图形信号中删除不构成非间行图形的单个场，从而产生来自输入图形信号并具有较低图形速率的输出图形信号的步骤；

通过修改输出图形信号的图形序列产生第一处理图形串的步骤；
和

把一个预置图形插入到第一处理图形串中以产生第二处理图形串的步骤。

18、如权利要求17的图形信号编码方法，其中，用于图形处理的辅助信息被附加到所述第二处理图形串上。

19、如权利要求17的图形信号编码方法，其中，所述的辅助信息包括：

第一标记，用于规定不构成所述非间行图形的单个场已被删除；
和

第二标记，用于规定在编码图形显示过程中，首先被显示的是第一场还是第二场。

20、如权利要求17的图形信号编码方法，其中，通过把原始24图形帧图形转换成每秒25帧视频信号来获得所述输入图形信号。

21、一种图形信号编码装置，包括：

检测装置，用于检测在输入图形信号中不构成非间行图形的单个场；

单个场删除装置，用于从所述的输入图形信号中删除检测到的

单个场；和

编码装置，用于根据没有单个场的顺序构成输入图形信号的图形串并对该图形串进行编码。

22、如权利要求21的图形编码装置，其中，所述的检测装置检测在通过编辑将第一场序列和第一场序列耦合到一起的过程中在一个编辑点处产生的不构成非间行图形的单个场，所述第一序列是通过把所述输入图形信号中的一个非间行图形转换成从一个间行图形的第一场开始的两个场获得的，并且所述第二序列是通过把所述输入图形信号中的一个非间行图形转换成从所述间行图形第二场开始的两个场获得的。

23、如权利要求22的图形信号编码装置，其中，所述的检测装置包括用于检查在从当前场向未来场延伸的场序列中的当前场是否是一个不构成非间行图形的单个场的装置。

24、一种图形信号编码装置，包括：

检测装置，用于检测输入图形信号中不构成非间行图形的单个场；和

重新构成装置，用于响应所述检测装置的输出重新构成所述输入图形信号的图形串并输出该重新构成的图形串。

25、如权利要求24的图形信号编码装置，其中，所述检测装置检测在通过编辑将第一场序列和第一场序列耦合到一起的过程中在一个编辑点处产生的不构成非间行图形的单个场，所述第一序列是通过把所述输入图形信号中一个非间行图形转换成从一个间行图形的第一场开始的两个场而获得的，所述第二序列是通过把所述输入图形信号中的一个非间行图形转换成从所述间行图形的第二场开始

的两个场而获得的。

26、如权利要求24的图形信号编码装置，包括：

删除装置，用于删除从所述输入图形信号中检测到的单个场；

所述的重新构成装置根据来自所述删除装置的没有单个场的场序列重新构成输入图形信号的图形串，并对该重新构成的图形串进行编码。

27、一种用于对具有构成非间行图形的第一和第二场不规则结合的输入图形信号进行编码的图形信号编码装置，包括：

检查装置，用于检查在输入图形信号中的一个场和相邻场之间的相关性；

存贮装置，用于存贮检测结果；和

判定装置，用于根据所存贮的检查结果判定由输入图形信号图形串的非间行图形构成的场的结合。

28、一种用于对具有构成非间行图形的第一和第二场不规则结合的输入图形信号进行编码的图形信号编码装置，包括：

用于从输入图形信号中删除不能构成非间行图形的单个场，从而产生来自输入图形信号并具有较低图形速率的输出图形信号的装置；

用于通过对输出图形信号的图形序列进行修改以产生第一处理后图形串的装置；和

用于把一个预置图形插入到第一处理图形串中以产生第二处理图形串的装置。

29、一种在其上记录有通过利用编码将第一场序列和第二场序列耦合到一起所产生的编码图形信号的记录介质，所述第一序列是

通过把所述输入图形信号中的一个非间行图形转换成从一个间行图形为第一场开始的两个场获得的，所述第二序列是通过把所述输入图形信号中的一个非间行图形转换成从所述间行图形第二场开始的两个场获得的，其中根据图形输入信号检测从第一场或第二场中的哪个场开始与所述非间行图形之一相对应的场图形组，根据检测的结果，输入图形信号的图形串被重新构成非间行图形串。

30、一种图形信号译码方法，包括如下步骤：

对在删除了不能构成非间行图形的单个场之后对图形信号进行编码所获得的信号进行译码：和

根据来自所述译码装置的信号，在单个场被删除的位置处插入一个场图形。

31、一种图形信号译码装置，包括：

用于对在删除了不构成非间行图形的单个场之后对图形信号进行编码所获得的信号进行译码的装置，和

用于根据来自所述译码装置的信号，在单个场被删除的位置处插入一个场图形的装置。

32、如权利要求31的图形信号译码装置，其中，所述被编码的信号包括：

第一标记，用于在每个编码帧的标题内规定不构成所述非间行图形的单个场已被删除；和

第二标记，用于规定在显示编码图形的过程中首先显示第一场还是首先显示第二场；和其中：

所述插入装置接收第一标记和第二标记，并当第一标记为1时，插入一个由第二标记所规定的奇偶检查场。

说明书

图形信号编码译码方法和装置 及信号记录介质

本发明涉及到一种用于对运动图形信号的记录和传送进行编码的图形信号编码方法和装置，用于在其上记录编码图形信号的信号记录介质和用于对编码图形信号进行译码的图形信号译码方法和装置。

在诸如是电信会议系统或电视电话系统的用于将运动图形信号传送给远方的系统中，使用图形信号的行相关或帧间相关来压缩图形信号的编码。当在诸如光磁盘或磁带的记录介质上记录运动图形信号并再现所记录的信号以便在一个显示单元上进行显示时，利用图形信号的行相关或帧间相关高效压缩图形信号的编码，以改善在所述记录介质上的记录效率。

就是说，如果希望在一个具有较小尺寸和较小记录容量的记录介质上记录具有大量信息的数字音频信号从而延长记录时间，就必须提供一种装置，以用于对视频信号进行高效编码和记录并对读出的信号进行高效译码。因此，已提出了一种响应这些请求的、由运动图形专家组(MPEG)系统例证的并利用视频信号相关的高效编码系统。

图1示出了使用MPEG系统对运动图形信号进行编码和译码的一

个系统结构的现有技术的例子。在图1中，利用扫描转换器152将由录像机(VTR)151提供的场信号转换成由编码器153进行编码的帧图形。利用MPEG系统，获得视频信号的帧间差以降低沿时间轴的冗余。然后，使用诸如离散余弦变换(DCT)的正交变换技术来降低沿空间轴的冗余，借此以对视频信号进行有效编码。编码数的信息可以被记录在记录介质154上。

为了对其上记录有高效编码信号的记录介质进行再现，通过译码单元155利用例如反正交变换对再现信号进行处理，以选定一个帧图形，然后，利用扫描转换器156对其进行转换以在监视器157上进行显示。

假定由录像机151提供有一个利用使用所谓的2-2下拉的(pull down)的电视电影技术，从一个胶片图形中获得的胶片图形。

所述2-2下拉技术是在根据所谓的逐行倒相系统(PAL)把每秒24个帧图形转换成每秒25帧或每秒50场的视频信号中广泛使用的电视电影技术。该系统通过隔行扫描以两个视频场读出胶片的每一个图形帧。

由于如此读出的两个场是从同一个图形帧中读出的，所以，被转换成一个帧结构的两个场可以被处理成非间行(non-interlaced)帧。即：这个帧等效于通过读出具有利用非间行扫描得到的视频帧的胶片的图形胶片所获得的一个帧。

通常，非间行帧在垂直方向上的逐行相关性要高于间行帧，因此，它的冗余和帧编码效率要高于间行帧。

如果对一个胶片的所有图象帧有规则地执行2:2下拉技术，那么，输入给编码器3的帧有必要是非间行帧，从而使得帧编码效率

较高，所以就不会存在问题。

如果使用传统的编码系统对通过所谓的电视电影操作利用间行图形从非间行图形转换而来的图形串进行诸如编辑的处理时，其编码效率趋向于降低。即，如果VTR151提供了由于诸如场编辑的后续处理所引起的含有不规则2:2下拉模式的视频信号，那么，输入给编码器153的帧就不必是非间行帧，所以，利用传统的编码器，帧编码效率就会如所不希望的那样被降低。有关这点将参照图2和图3加以解释。

在图2A中，利用电视电影处理的2:2下拉技术，将多个诸如是运动图形的图形帧的非间行图形NF转换成例如是所谓PAL系统的间行图形。利用所谓电视电影操作将图2A所示的非间行胶片图形序列转换成其每一帧都由一第一场(top_field)开始的间行图形的序列，如图2B所示。即：图2B所示的序列就是其中的每一个都是由第一场Ft和第二场Fb构成并以这个序列排列的多个帧CF的序列。相反，图2C所示非间行图形NF的序列被转换成其中的每一帧都开始于第二帧(底部帧)的间行图形的序列，如图2C所示。即：图2D所示的序列就是其中的每一个都由第二帧Fb和第一帧Ft构成并以这个序列排列的多个帧CF的序列。

如果这些序列在所示的编辑点 T_{E1} 、 T_{E2} 被相互结合到一起，就会产生一个不规则的序列，该不规则的序列会使与在原始图形中非间行图形NF的序列相应的配对场产生混乱，为图2E所示。在图2E所示的例子中，在所述编辑点 T_E 之后直接产生单个场 F_{x0} 。

图2E所示的序列优先影响图形编码效率，下面结合图3加以解释。

在编码期间，图2E所示的不规则序列被编码成图3A所示的组合矩形帧CP的帧。若被编码的帧对应于构成原始图形组合CP₁的非间行帧的组合CP₁，那么，如图3B所示，编码效率是高的。相反，利用除非间行图形NF正确组合以外的组合CP₂，如图3C所示，所述图形就会在其边缘部分包含有极高的频率成份，尽管原始图形是一个非间行图形，这样就会降低编码效率。在图3B和3C中，在表示一个实圆的图形沿横向移动的情况下，编码效率是降低的。

作为如上所述对利用电视电影处理所产生图形进行有效编码的技术，已提出了一种方法，包括从2:3下拉图形中删除重复场，然后构成多个帧，从而使得输入帧变成非间行帧。但是，由不规则2:2下拉技术如上述所产生的不构成非间行帧的单个场不同于由2:3下拉技术所产生的间行帧，因此使用这种技术不能解决问题。

根据上述，本发明即使是在需要编码的运动图形由于构成非间行帧的配对场的不规则周期而受到损害或作为执行非间行帧的序列的诸如场编辑的处理的结果，在包含有不构成非间行帧的单个场的情况下也能实现高效图形编码。由此，本发明的目的就是要提供一种图形编码方法和装置，借此，通过对间行图形串进行编辑，能够在即使产生不规则模式的情况下实现高频编码。

本发明的另一个目的就是要提供一种其上记录有如此编码的信号的信号记录介质和可以对所述编码信号进行有效译码的图形译码方法和装置。

根据本发明，检测在输入图形信号中不构成非间行图形的单个场并从所述输入图形信号中将其删除。输入图形信号的图形串是根据与所述单个场无关的场序列构成并被译码的。

根据本发明，当通过编辑将利用把在输入图信号中的一个非间行图形转换成由一个间行图形的第一场开始的两个场而获得的第一场序列和利用把在该输入图形信号中的一个非间行图形转换成由所述间行图形的第二场开始的两个场而获得的第二场序列相互耦合到一起时，可以从该输入图形信号中检测出上述的单个场来，从第一场或第二场的这个单个场开始的是一系列与可述非间行图形之一相对应的场图形，并且根据这个检测结果，所述输入图形信号的图形串被重新构成非间行图形串。

通过把具有作为一个基准场的一个予置场的两组场图形的场相关性加以比较并把一组具有较大相关性的场图形判断为与规则非间行图形相对应的场对，可以实现对从第一或第二场开始建立场图形的检测。在这种情况下，最好根据第一场的第 i 行和第二场第 i 行之间的相关性或根据第二场的第 j 行和第一场的第 j 行之间的相关性来获得配对场的相关性，以根据所述的相关性来检测所述的单个场。另外，最好根据多个场的相关性来检查当前场，以看看它是否是一个单个场，所述多个场相对于当前场来讲是过去场和未来场。

响应检测输出，输出一个用于控制编码的标记，该标记示出与输入图形信号图形串的非间行图形相对应的一组两个场图形是从第一场开始的还是从第二场开始的。

通过检测与非间行图形相对应的一组两个场图形是从第一个场还是从第二场开始的，并根据检测结果在编码期间重新构成一组场图形，可以实现编码，在其中，所述与原始非间行图形对应的一组两个场图形对应于用于编码的一个帧。

图1的简要方框图示出了一个传统运动图形编码/译码系统的配

置。

图2A、2B、2C、2D和2E示出了由于对两串非间行图形进行编辑所产生的不规则序列。

图3A、3B和3C示出了由不规则序列模式所引起的编码效率的降低。

图4示出了根据本发明一个实施例用于运动图形信号的编码/译码装置。

图5的方框图示出了一个用于检测和删除不构成非间行图形的单个场的例子。

图6示出了位于包含在相相对于是第一场的一个单个场是未来序列中的各个点处的信号和标记。

图7示出了位于包含在相相对于是第二场的一个单个场是过去序列中的各个点处的信号和标记。

图8示出了位于包含在相相对于是第二场的一个单个场是未来序列中的各点处的信号和标记。

图9示出了位于包含在相相对于是第二场的一个单个场是过去序列中的各点处的信号和标记。

图10示出了利用本发明的实施例检测和删除单个场的操作流程。

图11的流程图是图7流程的延续。

图12示出了另一个用于检测和删除不构成非间行图形的单个场的电路的实施例。

图13示出了一个用于检测与正确非间行图形相对应的结合的算法例。

图14的电路方框图示出了一个用于检测与正确非间行图形相对应的结合的电路例。

图15示出了利用图9所示的结构重新构成与单个场的输入、检测和删除相关的输入帧的例子。

图16示出了利用图9所示的结构重新构成与单个场的检测、删除和输入相关的输入帧的另一个例子。

图17示出了用于判断构成非间行帧的两个场的结合的算法的例子。

图18的流程图示出了与图14所示的算法相对应的处理序列。

图19示出了一个根据本发明的图形信号编码装置。

图20示出了一个不规则模式，在这种模式下，非间行图形的序列的结合受到了扰乱。

图21A、21B和21C示出了一个检测规则模式的操作。

图22A、22B和22D示出了不适宜模式的编码。

图23示出了利用重复在前场内插一个单个场的例子。

图24示出了通过重复在前场内插一个单个场的例子。

图25示出了一个图形组的例子。

图26的流程示出了根据本发明的一个修改检测和删除一个单个场的操作。

图27A和27B示出了在包含在相相对于是第一场的单个场是未来序列内的各部分中的信号和标记。

图28A和28B示出了在包含在相相对于是第一场的单个场是过去序列内的各部分中的信号和标记。

图29示出了在本发明另一实施例中，在一个标记规定删除一个

单个场的情况下各部分内的信号和标记。

图30A、30B和30C示出了在图形类型和用于指定与编码器相关的标记的方法之间的关系。

图31示出了重新构成用于检测、删除和输入单个场的一个输入帧的例子。

图32示出了重新构成用于检测、删除和输入单个场的一个输入帧的另一个例子。

图33示出了用于判断组成非间行帧的两个场的结合的算法的另一个例子。

图34流程图示出了与图30所示算法相关的处理序列。

图35示出了与结合图3示出的编码位流相关的图形输出信号和标记。

图36示出了与结合图4示出的编码位流相关的图形输出信号和标记。

图37示出了与结合图5示出的编码位流相关的图形输出信号和标记。

图38示出了与结合图6示出的编码位流相关的图形输出信号和标记。

图39示出了一个图形编码单元的结构。

图40A和40B示出了高效编码的原理。

图41A和41B示出了与压缩图形数据相关的图形类型。

图42A和42B示出了对运动图形信号进行编码的原理。

图43的方框图示出了根据本发明的图形信号编码器和译码器的结构的例子。

图44示出了图40所示的格式转换电路17的格式转换操作。

图45示出了图40中编码器18的结构例。

图46示出了图42预测转换电路52的操作。

图47示出了图42所示转换电路55的操作。

图48A、48B和48C的方框图示出了图40中译码器31的结构例。

下面参照附图详细描述本发明的最佳实施例。

图4简要示出了作为本发明一个实施例的用于图形信号的图形信号编码和译码系统。

在该图中，一个输入运动图形信号S1被从例如一个录像机(VTR)161提供给输入端171。这个输入运动图形信号S1是一个包含有单个场的运动图形信号，所述单个场不能构成非间行帧，或者由于此非间行帧序列执行例如是场编辑的处理而使所述单个场在非间行帧配对场的周期内处于无规则状态(图2)。

输入运动图形信号S1被提供给非间行帧检测电路172。非间行帧检测电路172检测组成非间行帧的第一和第二场的结合或者是检测不构成非间行帧的单个场。对被如此检测到的单个帧进行控制，以使其不被下游(downstream)侧的编码器163所编码。这一点可以通过利用非间行帧检测电路172从所述输入运动图形信号中删除单个帧来实现，后面还要对此加以解释。

非间行帧检测电路172输出一个场信号S6，扫描转换器162按输入序列将场信号S6转换成帧图形，以产生一个帧信号，该帧信号被提供给编码器163，有关编码器163，下面还将详加解释。在输出端173得到由编码器163输出的编码位流信号S204，并将其提供给和存贮在例如存贮介质164中。

来自例如存贮介质164的编码后位流信号被提供给译码装置的输入端176，而后提供给译码器165，从而将其译码成帧信号206。这个帧信号206被提供给扫描转换器166，以将其转换成场信号S207，场信号S207被提供给输出控制器177，在这里，它变成一个输出运动图形信号S208。该输出运动图形信号S208在输出端178被取出，以在监视器167上进行显示。这一部份将在下面详加解释。

下面结合图5来解释被构成用于检测单个场并随后删除所检测到的单个场的如图4所示的非间行帧检测电路172的结构。

提供给图5输入端200的是图4的输入运动图形信号S1，该信号随后被存贮在场存贮器201到203中。这样，当所述当前场被存入存贮器202时，过去场和未来场被分别存贮在存贮器201和203中，加法器204计算在过去场信号S301和当前场信号S302之间逐个象素的差，以获得差值信号S304。利用绝对值电路206获得差值信号S304的绝对值，并利用求和电路208求出每个场绝对值的以场为基础的和。类似地，加法器205获得在未来场信号S303和当前场信号S302之间象素基础上的差值，以获得差值信号S305。利用绝对值电路207获得差值信号S305的绝对值，并利用求和电路209计算每一场绝对值的以场为基础的和。和信号S308和S309被输入给比较器201，比较器201将信号S308和S309的幅值加以比较，并将比较结果输出给标记S2。所述标记S2的值是：

当 $S308 \geq S309$ 时， $S2 = 1$ 和

当 $S308 < S309$ 时， $S2 = 0$

如果 $S2 = 1$ ，它表示当前场与未来场更相关，若 $S2 = 0$ ，它规定当前场与过去场更相关。直到过去两个场在S2的值被存贮在延时电

路211和212中。

图6到图9示出了计算实际输入运动图形信号S1的标记S2的例子。在图8和图9中，大写字母和小写字母分别用于规定第一场(top_field)和第二场(bottom_field)。一个非间行帧由相同的大字和小写字母构成。符号“?”表示一个不构成非间行帧的单个帧。从图6到图9可以看出，如果一个非间行帧被转换成两个从一个间行帧的第一场(top_field)开始的场如图6的|Aa|所示，那么，所述的第一场(top_field)就更与未来的第二场(bottom_field)相关，从而使得标记S2为1。另一方面，第二场(bottom_field)与过去的第二场(bottom_field)更加相关，所以，S2为0。如果一个非间行帧被转换成从一个间行帧第二场(bottom_field)开始的两个场，如图6 d | D所示，那么，第二场(bottom_field)就更与未来的第一场(top_field)相关，所以，标记S2是1。另一方面，所述第一场(top_field)和过去第二场(bottom_field)更加相关，所以，标记S2 = 0。

下面来解释在一个单个场“?”处的标记S2。由于在图6和图8中的单个场被包括在某个未来序列中，可以标记S2被设定为1。而在图7和图9中，所述单个场被包含在某从此过去序列中，所以标记S2被设定为0。

根据标记S2利用图5所示的控制电路215检测单个场并利用场删除单元218从输入运动图形信号S1中删除检测到的单个场。

所述算法的流程示于图10和图11。其中， f_p 表示从输入所述输入运动图形信号的场开始的场时间，且场时间 f_p 增加表示信号S1的时间朝着未来发展。例如， $f_p = f_p + 2$ 表示信号S1的时间朝着未发前进了两个场，即S1的输入被超前了两个场。诸如对于S2的后缀 $[f_p]$

表示场时间。

当检测到一个不构成非间行帧的单个场时，在一个场时间处信号S3变为1。

在时间 f_p 处当前场的奇偶性是由信号S4反映出来的，对于第一场(top_field)，S4变成1，而对于第二场(bottom_field)，S4变为0。

信号S5是一个标记，它表示自时间 f_p 开始直到在所述时间 f_p 未来侧上的两个场为止，场已经被删除。在S3处检测到的单个场被删除。

信号S5可以被认为是一个标记，用于规定与它最接近的两个过去场或最接近的两个未来场与被删除的场最相关。对于两个场信号S5为1则表示极度相关。与被删除场相邻的两个过去和未来场构成了被编码的帧。

信号S4可以被认为是一个标记，用于规定当对被编码的帧进行译码和再现时，要首先显示被编码帧第一场(top_field)或第二场(bottom_field)。

下面参考图10和11的流程图来解释单个场检测的操作。

在图20的第一步301，将信号S1的场时间 f_p 设定为1($f_p = 1$)。

在步骤302，时间 f_p 的场的奇偶性识别所述场是否是第一场(top_field)。若在步骤302检测的结果是“YES”，即假如所述场的奇偶判断是第一场(top_field)，处理通过节点303转向步骤305。若检测结果是“NO”，处理通过节点304转向步骤316。

在步骤305，鉴别朝向未来两个场的标记 $S2[f_p + 2]$ 是否是1，若其结果是“YES”，处理进到步骤306，反之，则转向步骤309。

在步骤306, 信号 $S4[f_p]$ 和信号 $S5[f_p]$ 被分别设定为1和0 ($S4[f_p] = 1, S5[f_p] = 0$).

在步骤307, 信号S1的时间向前了两个场 ($f_p = f_p + 2$), 此后, 处理通过节点303返回到步骤305。

在步骤309, 鉴定朝未来一个场一标记 $[f_p + 1]$ 是否是1。若鉴定结果是“YES”, 处理进入到步骤310, 反之则转向步骤311。

在步骤310, 时间 f_p 的场是一个不构成非间行帧的单个场。标记 $f_s[f_p]$ 是1, 并从信号S1中删除时间 f_p 的场。

在步骤311, 时间 $f_p + 2$ 的场是一个不构成非间行帧的单个场。标记 $S3[f_p + 2]$ 是1并从信号S1中删除时间 $(f_p + 2)$ 的场。

在步骤312, 紧接着步骤310和311, 标记 $S4[f_p]$ 和标记 $S5[f_p]$ 被设定为1 ($S4[f_p] = 1, S5[f_p] = 1$)。

在步骤313, S1的时间向前了三个场 ($f_p = f_p + 3$)。然后, 处理通过节点304返回到图8的步骤316。

在步骤316, 判断朝前两个场的标记 $S2[f_p + 2]$ 是否为1。若结果为“YES”, 处理进到步骤317。反之, 则转向320。

在步骤317, 标记 $S4[f_p]$ 被设定为0, 同时标记 $S5[f_p]$ 被设置为0 ($S4[f_p] = 0, S5[f_p] = 0$)。

在步骤320, 判断与朝前一个场相关的信号 $S2[f_p + 1]$ 是否是1。若答案是“YES”, 处理进入到步骤321, 反之则转向322。

在步骤321, 时间 f_p 的场是一个不构成非间行帧的单个场。标记 $S3[f_p]$ 是1, 且从信号S1中删除时间 f_p 的场。

在步骤322, 时间 $f_p + 2$ 的场是一个不构成非间行帧的单个场。标记 $S3[f_p + 2]$ 是1, 并从信号S1中删除时间 $(f_p + 2)$ 的场。

在步骤323, 当接着步骤321和322, 标记S4 $[f_p]$ 和标记S5 $[f_p]$ 被设定为1 (S4 $[f_p] = 0$, S5 $[f_p] = 1$)。

在步骤324, S1的时间被向前了三个场 ($f_p = f_p + 3$)。随后处理通过节点303返回到图10的步骤305。

利用这种方式, 可以检测和消除不构成非间行帧的单个场。

在图6到图9中, 除了标记S2以外, 还示出了来自单个场检测和删除电路172的标记S3到S5和场信号S6的例子。

参看图12, 解释了非间行帧检测电路172的一个修改电路。在图12中, 与图5所示的对应部分使用相同的标号。

图4所示的输入运动图形信号S1被提供给图12的输入端200。这个信号被序列地存贮在场存贮器201到203中。这样, 如果诸如是第n个场的当前场被存贮在存贮器202中, 那么作为一个过去场的第(n-1)个场和作为一个未来场的第(n+1)个场被分别存贮在存贮器201和203之中。

非间行帧计算电路224就第(n-1)个场S321和第n个场S322是否代表了非间行帧的结合进行计算, 并将计算结果S324输出给比较器226。非间行帧计算电路225就第n个场S322和第(n+1)个场S323是否代表了非间行帧的结合进行计算, 并将计算结果S325输出给比较器226。

下面结合图13来解释用于通过这些非间行帧计算电路224、225就两个给定场是否代表非间行帧的结合进行计算的算法。在图13中, 所述算法是以所谓的C语言表示的。

参看图13, “F像素(pixel), 行(line)” 分别表示在计算电路224和225中由第(n-1)个场S321和第n个场S322构成的帧中的像素

值和由第 n 个场S322和第 $(n+1)$ 个场S323构成的帧中的象素值。

“line(行)”表示一个帧的行数的计数值，该计数值可以高达最大值：“frame_vertical_size”。第0行，即“line = 0”，第1行，即“line = 1”分别表示第一场(top_fitld)和第二场(bottom_fitld)。“Pix”表示在每一行上的象素数的计数值，且该计数值可达最大值“frame_horizontal_size”。第0象素，即“Pix = 0”开始于每一行最左边的象素。

图13中的方程式a计算用于计算第一和第二场第 i 行之间的相关性的逐行差值“dif1”。如果每个场是由 N 行构成的， i 的值则部分或全部的为从0到 $(N-1)$ 。另一方面，方程式b计算用于计算第二场的 j 行和第一场的第 $(j+1)$ 行之间相关性的逐行差值“dif2”， j 的值为部分或全部从0到 $(N-1)$ 。利用等式c来计算逐行差值绝对值的和“Var”。

通过同时使用等式a和b，进行计算以检查任一给定的两个场是否代表了与非间行帧相关的结合。可以利用平方和的计算来代替与等式c的绝对值相关的计算。

图3所示算法可由图11所示的硬件表示。

参看图14，计算两个相邻行差值绝对值的和Var1、Var2以用于场组 CF_1 、 CF_2 。即：对于场组 CF_1 ，通过减法器111获得由图14中实线表示的第一场各行数据和由虚线表示的第二场各行数据之间的差值以及第二场各行数据和第一场各行数据之间的差值，利用绝对值电路112获得其对应的绝对值并利用加法器113获得其绝对值之和，从而获得各相邻行差值绝对值的和Var1。至于场组 CF_2 ，利用减法器121获得由图14中实线所示的第一场各行数据和由虚线所示的第

二场各行数据间的差值以及第二场各行数据和第一场各行数据之间的差值。另外，利用绝对值电路122获得其相应的绝对值并利用加法器123获得其绝对值之和，从而获得各相邻行差值绝对值的和Var 2。将绝对值之和Var1和Var2彼此加以比较，较小的和值被判断为表示与正确非间行帧相对应的配对场。给出这个判断的理由是正确非间行帧的配对场受靠近边缘的高频成份扰乱的程度较小。这里所使用的绝对值和可由平方和来取代，即可以使用平方电路来取代绝对值电路112和122。

在图13所示的算法中，计算电路224和225输出作为计算值“Var”的信号S324和S325。这两个信号被提供给比较器226，在比较器226中信号S324和S325彼此进行比较，并输出比较结果标记S22。当 $S324 < S325$ 时，标记S22的值为0，即 $S22 = 0$ ，当 $S324 \geq S325$ 时， $S22 = 1$ 。另一方面，假如 $S22 = "1"$ ，则它规定第n个场是第(n+1)个场和非间行帧的结合。这个算法的优点在于假如任一给定的两个场对应于非间行帧的结合，那么在两个场之间的逐行相关性是高的，从而使逐行差值绝对值的和变小。

图15和16示出了一个对与实际输入运动图形信号S1相关的标记S22进行计算的例子。如在图6到图9所示的例子一样，在图15和图16中，分别用大写字母和小写字母规定第一场(top_field)和第二场(bottom_field)。非间行帧是由相同的大写和小写字母构成的。符号“?”表示不构成非间行帧的单个场。

从图15和16中可以看到，如果一个非间行帧被如图6 | Aa | 处所示的转换成从一个间行帧第一场(top_field)开始的两个场，那么所述第一场(top_field)“A”就更与未来第二场(bottom_field)

“a”相关，所以，标记 $S22 = 1$ 。另一方面，第二场(bottom_field) “a”更与过去第二场(bottom_field) “A”相关，所以， $S22 = 0$ 。若一个非间行帧被如图6 d | D处所示的转换成从一个间行帧第二场(bottom_field)开始的两个场，那么所述第二场(bottom_field) “d”就更与未来第一场(top_field) “D”相关，所以标记 $S22 = 1$ ，另一方面，所述第一场(top_field) “D”更与过去第二场(bottom_field) “d”相关，可以标记 $S22 = 0$ 。

产生单个场“？”的原因在于在每个时间点处的非间行序列上在前执行了场编辑。

下面解释在单个场“？”处的标记 $S2$ 。由于在图15中的单个场包含在某个未来序列中，未来场更相关，因此 $S2 = 1$ 。另一方面，单个场被包含在某个过去序列中，它与所述过去场更相关，由此标记 $S22 = 0$ 。

虽然关于非间行帧的结合的判断是以比较器226的输出标记 $S22$ 为基础的，但有时会出现输出错误标记 $S22$ 的情况。可以通过把 $S22$ 的一系列值按时间序列存贮到存贮器或寄存器227中，并通过根据标记 $S22$ 多个值的阵列模式给出有关非间行帧结合的最终判断来减少所述的这种错误判断。

图17示出了一个用于根据标记 $S22$ 来判断非间行帧结合的例子，在该例中，所述算法利用所谓的C-语言表示。

在图17中， n 用于规定从输入一个输入运动图形信号 $S1$ 的一个场开始所经过的场时间， $S1[n]$ 和 $S22[n]$ 用于规定 $S1[n]$ 的标记 $S22$ 。若 n 增加，它表示信号 $S1$ 的时间向着未来前进。例如， $n = n + 2$ ，表示信号 $S1$ 的时间向着未来前进了两个场，即 $S1$ 的输入前进了两个场。

在图17中，变量“isTop_field”反映了作为初值 $n=0$ 的那个场的奇偶性。由于在图15和16中输入信号 S_1 的场序列开始于顶部场，所以，“isTop_field=1”，如图17中的第4行所示。由于在场时间 n 处的场奇偶性是反映在变量top_field_first[n]中的，所以，对于top_field或bottom_field，它被分别设置成“1”或“0”。变量repeat_first_field[n]是一个用于删除单个变量的标记。变量pattern[n]被用于控制图形输出控制器228(图12)的输出。

在满足图17中从第7行到第10行的AND条件的情况下，利用其值为“1”的repeat_first_field[n]命令删除单个场。根据“isTop_field”来设置变量top_first_field[n]。接着对表示为第29行 $n=n+3$ 的后三个场执行检查。“isTop_field”被取反，从而使其作为如第18行和第26行的指示的用于下一个检查场的奇偶性。

若上述的AND条件没有被满足，保持变量的先前状态。针对如第39行所指示的后两个场执行下一次检查。

图18示出了与图17算法相对应的算法。

在图18中，图17所示的条件 CND_1 到 CND_5 对应于图18的相应判断步骤 CND_1 到 CND_5 。图18中的其余步骤用于设定变量 n ，isTop_field、top_first_field[n]，repeat_first_field[n]和pattern[n]。

下面来解释根据图15和16所示的算法获得与图15和图16所示输入信号 S_1 相关的top_first_field[n]、repeat_first_field[n]和pattern[n]的例子。在图15和16中， S_{24} 和 S_{25} 分别用于规定top_first_field[n]和repeat_first_field[n]。

下面解释图12的图形输出控制器228控制显示单元229输出图形输出信号 S_{26} 的处理。延迟单元229是一个用于存贮位于相对于第 $(n$

+ 1)个场的未来一侧上的场的存贮器并能够输出具有由图形输出控制器228所规定延迟的场图形。在图15和16的例子中，需要一个7场存贮器。帧存贮器可以被用于构成延迟单元229和场存贮器202到203。

在变量“pattern”的基础上，可以输出下面(1)到(6)所示的图形输出S26。图15和16示出了所规定的例子。

(1): 对于pattern = 1

连续输出被从S1开始延迟4个帧的场信号S104的两个场。

(2): 对于pattern = 2

输出被从S1开始延迟三个场的信号S103。然后以下述序列:

即信号S104和两个无效场 | - - | 被输出。

(3): 对于pattern = 3

按下述序列，即信号S104的两个场和两个无效场被输出。

(4): 对于pattern = 4

输出信号S104，然后输出从信号S1开始延迟了5个帧的信号S105。

(5): 对于pattern = 5

连续输出信号S104的两个场。

(6): 对于pattern = 6

顺序地输出信号S104和信号S105。

该无效场是输入给下一级编码器但不被有效处理的场。这样一个无效场使用符号“-”加以规定。在图15和16中，示出了与输入信号S1相关的输出S26的帧延迟量。

图15和16的例子表明，和输入信号S1相比较，输出S26与单个场“?”无关，同时构成非间行帧的第一和第二场变为一间行帧，

其中是第一场首先输入的。通过插入无效帧，S26的帧周期被根据输入S1的帧周期加以锁定。这样，就可以在根据输入S1的帧周期的操作中锁定下游侧的编码器。

参看图4，扫描转换器162按照输入序列将由非间行帧检测单元172输出的场信号S2转换成一个帧图形。在图15和16所示的例子中，S26的“1”表示所述帧边界。由此，可以看到在根据上述模式输出的两个场间隔处构成了一个帧图形。

输入帧S203，相关标记S24(top_first_field[n])和repeat_first_field[n])以及无效帧标记被输入给编码器163。使用与在MPEG2中定义的相同名称的标记可以对标记S24(top_first_field[n])和repeat_first_field[n])进行编码。编码器不对已经设定了无效帧标记的输入帧进行有效处理。后面将要对编码器163的详细结构进行解释。

一个编码位流S204被提供给记录介质164。在上述的实施例中，使用7个S22信号，即S2[n]到S2[n+6]信号的阵列模式构成用于判断图23所示的非间行帧的结合的算法。但是，信号S22更大量的阵列模式被用于进一步减少对非间行帧给出错误判断的出现。相反，信号S22较小量的阵列模式被用于以增加对非间行帧给出错误判断的可能性为代价的情况下减少输出S26的帧延迟量。

参看图19，详细解释根据本发明的所述图形信号编码装置的一个修改实施例。

作为一个输入图形信号提供给图19输入端101的包括：为图2B所示的由把所述非间行图形NF转换成多个场组，每个场组由两个场图形、即帧CF构成，每个帧由一个间行图形的第一场或顶部场开始

而获得的一个图形串，以及如图2B所示通过把非间行图形NF转换成多个帧组，每个组由两个场图形，即帧CF、且每个帧开始于一个间行图形的第二帧或底部帧而获得的另一图形串。通过图2D所示的编辑将这两个图形串链接起来。

输入图形信号被提供给作为用于预处理电路102的检测装置的不规则模式检测电路103。在这里判断与非间行图形之一相对应的两个场是开始于第一场还是开始于第二场。不规则模式检测电路103输出的不规则模式检测信号和图形数据被提供给一个作为重新构成装置并用于非间行图形的重新构成电路104。根据对应于输入图形信号图形串非间行图形的配对场图形是开始于第一场或是第二场，这个重新构成电路104重新构成并输出一个非间行图形。

用于非间行图形的重新构成电路104的输出信号作为预处理电路的输出信号提供给编码器105。在当前修改实施例中，通过利用图形信号逐行相关性或逐帧相关性执行压缩编码的编码器被用作编码器105，除了图形数据以外，作为由预处理单元向编码器105提供的信号，还使用了后面将要解释的三种信号，即标记：`picture_coding_type`、`top_first_field[n]`和`repeat_first_field[n]`。编码器105根据这些标记产生编码数据并在输出端106输出所产生的数据。下面来解释编码器105的一个例子。

首先，解释利用作为图19预处理电路102的检测装置的不规则模式检测电路103对构成一个正确非间行图形的配对场帧进行检测的方式。

图20示出了一种不规则模式，在这种模式下，非间行帧的序列组合受到破坏。可以看到当通过编辑将与从第一场Ft开始的一个非

间行图形相应的配对场的序列和与从第二场 Fb 开始的一个非间行图形相应的配对场(帧)的序列相互链接到一起时,就会产生不适宜的序列 $SQ1$ 或 $SQ2$ 。

由此,如图21A所示,就形成了包括有一个预置场,在这里是第二场 Fb_0 在内的两组场 CF_1 和 CF_2 。所述场或帧中的一个 CF_1 由一对场,即第一场 Ft_1 和第二场 Fb_0 组成,并按这个序列排列,而这些场或帧中的另一个 CF_2 是由配对场,即第二场 Fb_0 和第一场 Ft_2 组成的。这些场组 CF_1 或 CF_2 中的一个表示与所述正确非间行帧相应的结合,如图21B或21C所示。

为了检测构成正确非间行图形的帧的场组,可以使用图14所示的检测电路。

下面解释在非间行序列的结合受到破坏的情况下,用于在不规则模式检测之后对不适宜模式进行编码的方法。

图22A示出了一个如上所述检测的不规则模式或不适宜模式 SQ 。在对序列编码中所述的不适宜模式就是如图22B所示的单个场 Fx 。单个场 Fx 是一个在编辑中产生并在形成非间行帧对过程中没有配对物的场。如果使用例如帧相关性进行编码,那么,这个单个场 Fx 就不能被编码。所以,如图22B所示,对于使用例如朝前或朝后侧场的内插法而言,这样一个单个场 Fx 基本上被删除。

图22C示出了一种情况,在这种情况下,通过重复在前场 Ft_1 而不是删除单个场 Fx 来执行所述的内插法。组成这样一个在前帧 Ft_1 重复的场图形组称之为 Cft_p 。如图22D所示,还可以通过重复后继场 Ft_2 以取代删除单个场 Fx 来执行这种插入。包括这种重复后继场 Ft_2 的场图形组称之为 Cft_b 。在图22C和22D中,在与被编码的与非间

行图形成帧相对应的配对场组中，分别称从第一场 F_t 开始的场图形组和从第二场 F_b 开始的场图形组为 C_{F_t} 和 C_{F_b} 。

图23示出了一个例子，在这个例子中，利用使用后面将要解释的帧相关性的运动图形编码对图22C所示的模式进行处理。在这个编码中，使用两种标记的结合，即标记 $top_first_field[n]$ 和 $repeat_first_field[n]$ 的结合可以与这个非间行图形相对应的场图形组 C_F 或帧的序列进行编码。

在这些标记中，当被编码的场图形组成帧是开始于第一场 F_t 的图形组 C_{F_t} 或开始于第二场 F_b 的图形组 F_t 时，标记 $top_first_field[n]$ 是1或0。

当所述被编码的场图形组成帧的第一场分别被重复或不重复显示时， $repeat_first_field[n]$ 是1或0。

这样，通过将图23所示的数字值指定给所述标记，就可以实现编码。

图24示出了对图22D所示模式进行编码的例子。

在图23和24中，对于多组编码帧或一组场图形 C_{F_t} 和 C_{F_b} ，两个标记，即 $top_first_field[n]$ ， $repeat_first_field[n]$ 都是1。其中，所述开始场是第一场 F_t ，且这个开始场被重复显示。应当注意，在图24所示的场图形组 C_{F_t} 中，其中的后向场 F_t 被重复，作为开始场的第一场 F_t 的数据被重复显示以作为编码数据。

用于图19所示非间行图形的重新构成电路执行包含有下面将要解释的图形类型安排的编码帧 CF 或场图形组的重新构成处理，并输出三种标记，即： $picture_coding_type$ ， $top_first_field[n]$ 和 $repeat_first_field[n]$ 。这些标记与图形数据一起提供给编码器

105。

编码器105使用上述标记通过下述方法产生编码数据，并在输出端1-6输出数据。

来自输出端106的编码输出信号被记录在诸如是一个光磁盘的信号记录介质上，用于产生其上被记录有利用高效编码进行编码的信号的记录介质。

利用上述运动图形记录方法和装置，可以在构成一个帧之后执行编码，以使得即使是由输入运动图形非间行帧构成的场对的周期是不规则的或者是在输入运动图形中含有不构成非间行帧的单个场，也能够使输入帧变成非间行帧，从而实现高效图形编码。当对包含有由于以25 HZ的帧速率以2:2下拉技术从胶片图形所产生的视频信号进行例如场编辑的处理所获得的不规则2:2下拉模式的视频信号进行编码时，上述的方法或装置是有效的。

下面解释作为编码标准的MPEG编码标准的使用。

利用MPEG系统的编码，每个视频序列被分成多个图形组(GOP)，即诸如12帧的帧组。利用预测系统，每个帧被分类成三组，即：帧内编码图形成I-图形，预测编码图形成P-图形，其中运动预测来自根据过去和未来的P-或I-图形，以及双向编码图形成B-图形，其中，运动预测来自根据过去和未来P-或I-图形。在本实施例中，当对作为一个B-图形输入帧进行编码时，控制结合图4到图11所解释的标记S5以使其不等于1。用于非间行帧检测电路172的控制方法在这方面不同于图4所示的方法，尽管在其余方面两种方法是很相似的。

下面参照图26所示的流程示来解释利用MPEG系统作为一个编码

例在非间行帧检测电路中所作处理。信号S7规定所述帧编码方法，并由编码单元163指定。图26的算法与图10和图11所示算法之间的区别在于对到步骤S331和S332的条件转移305的条件作了修改。由于图26所示的算法的其余部分与图10和图11所示的算法相同，所以，只示出了包括有步骤S331、S332的修改部分而没有示出其余的部分。

就是说，图10所示的条件转移步骤305被修改成图26所示的条件转移步骤S331。在这个步骤S331，它要识别朝前了两个场 $S2[f_p + 2]$ 的标记是否等于1或识别信号S7是否规定了一个B-图形。另一方面，图11的条件转移步骤316被修改成图26所示条件转移步骤332。在步骤332，要识别朝前了两个场 $S2[f_p + 2]$ 的标记是否等于1或识别信号S7是否规定了B-图形。

示出在各部分处的信号和标记的例子的图27A和27B被直接用于使用信号top_field去删除单个场。图27A和27B使用图6到11的方法和图23所解释的方法。

所述场“?”的时间是当前时间 f_p 。在图27A中，“?”被检测为单个场并予删除。帧|fF|被指定为B-图形，标记S5被设定为1。

在图27B中，由于B-图形的编码被规定为在场“?”的时间 f_p 处进行，所以，“?”不被检测为单个场。当场时间向前前进了两个场($f_p = f_p + 2$)时，规定P-图形的编码。场“F”被检测为单个场并被删除。帧|gG|被编码成P-图形，标记S5被设定。假如在图26所示的算法中，场“F”的时间被判断为当前时间 f_p ，判断场“F”是一个在当前时间 f_p 的暂时朝前侧上的场序列中不构成非间行帧的一个帧。

删除单个场的另一个例子示于图28A和图28B。在这个例子中，利用信号bottom_field来删除单个场。图28A和28B示出一个图6到11解释的方法例如图26解释的方法例。

在图28A和28B中，场“C”的时间是当前时间 f_p 。在图28A中，“?”被检测为单个场并被删除。帧|cC|被指定为B-图形，标记S5被设定为1。

在图28B中，与场“C”相关的时间是时间 f_p ，且编码方法被指定给B-图形，所以，“?”不被检测为单个场。当场时间向前了四个场时，编码方法被指定给P-图形，场“d”被检测为单个场并被删除。帧|Ee|被编码成一个P-图形，标记S5被设置为1。如果在图26所示的算法中与场“d”相关的时间是当前时间 f_p ，那么，在当前时间 f_p 暂时朝前侧上的场序列中的场“d”被判断为是一个不构成非间行帧的单个场。

利用上述过程，检测并删除不构成非间行帧的单个场。

这样，与单个场无关并由图4所示场检测和删除电路172输出的场信号S6被利用扫描转换器162转换成依输入序列的帧图形。下面结合图27和28来解释信号S6的一个例子，输入帧信号S203和相关标记S4和S5被输入给图形编码器163。可以使用在MPEG2中定义的top_first_field[n]和repeat_first_field[n]对两个标记S4和S5进行编码。下面将要描述编码器163的详细结构。编码位流S204被提供给记录介质164。

当图形译码器具有MPEG2所谓MP@ML(主等级处的主外形)级别的性能时，这里所解释的实施例是必不可少的。就是说，当使用MPEG2的repeat_first_field[n]来传送表示在前已经删除了不构成非间

行帧的单个场的一个标记时，对于25 HZ的帧速率，MPEG2的MPML禁止repeat_first_field[n]被设定为1。由此，上述特定的实施例被用于避免这个规则并实现高效图形编码。

下面来解释针对标记S5 = 1的情况下必须把编码方法设定给I或P的实施例。

通常，在GOP，例如在图25所示的GOP结构中，P-图形或I-图形是周期性出现的。每三个帧都要出现多个P-图形或I-图形。

在当前实施例中，当结合图6到11所解释的标记S5被设置为1时，必须将帧编码方法设定给一个I-帧或P-帧。这样，当前实施例在控制用于规定由编码器163执行的帧编码方法的信号S7的方法方面与先前的实施例不相同。当前实施例的其余方面与结合图6到11解释的实施例相同。

下面结合图29和4来解释一个示意例。场“？”的时间被检测为单个场并被删除，同时，标记S5被设定为1。编码器163对其作出响应，并利用信号S7指定所述P-图形。应当注意，如果每三个帧使用P-图形，那么，B-图形将被规定。

利用扫描转换器162按着输入序列对由非间行帧检测电路172输出的场信号S6进行转换。图29示出了信号S6的一个例子。

输入帧信号S203和相关标记S4和S5被输入给图形编码器163。使用MPEG2中定义的top_first_field[n]和repeat_first_field[n]可以对两个标记S4和S5进行编码。后面还要描述编码器163的详细结构。编码位流S204被提供给记录介质164。

当图形编码器具有MPEG2所谓MP@ML(高级处的主外形)级别的性能时，这里所解释的特定实施例再一次成为必不可少的。即：禁止

把B-图形的repeat_first_field[n]如上所述设置为1的规则可以被避免并实现高效编码。

同时，如果上述MPEG编码标准用于PAL(逐行倒相)系统的图形信号，那么就要规定利用B-图形去禁止在PAL系统序列中的场重复。下面结合图30来解释在上述PAL系统中出现的如上所述的不规则电视电影模式和待被适当处理的某些需要。

在图30中，所述编码器决定是否要对不适宜的不规则模式进行场重复。在当前实施例中，在I-图形或在B-图形处安排执行场重复。

就是说，利用如图30A所示的包含有不规则模式的序列，通过安排含有场重复的场组CFT_n，可以在不违背上述规则的情况下进行编码，从而与所述的P-图形对应，如图30B所示。同时，如果在编码期间含有场重复的场组CFT_n对应于B-图形，那么，就会违背上述规则。因此要作安排以避免这种编码，且如此安排的序列被传送给图19的编码器105。

在非间行图形重新构成电路104中，执行包括有如上所述图形类型安排的场图形组DF编码帧的重新构成，并输出三种标记：picture_coding_type、top_first_field[n]和repeat_first_field[n]，且这些标记和图形数据一起被传送给编码器105。

下面详细解释在图12中构成的一个使用非间行帧检测电路的例子。

图33示出了使用图12所示电路根据标记S22对非间行帧的结合进行判断的算法的例子。第7行上的标记“picture_coding_type”表示由图4所示下游侧编码器163指定的帧编码方法。该算法就关于

在第7-11行上的AND(与)条件方面与图17所示的算法不同。在当前实施例中，如果附加的“picture_coding_type”不是B-图形，则与在图17算法中用于repeat_first_field[n]等于1的有关条件保持不变。

图34示出了与图33所示算法相应的流程。

在这个图中，图33中的条件CND₁-CND₅与图34中的条件识别步骤CND₁₁-CND₁₅对应。图34的其余步骤示出变量n, isTop_first, top_first_field[n], repeat_first_field[n]。

图31和32示出了一个例子，其中，图31示出了利用图33所示算法对与图15所示相同输入S1相关的top_first_field[n]和repeat_first_field[n]的识别结果，图32示出了利用图33所示算法对top_first_field、repeat_first_field和pattern进行判断的结果。

图31和图15之间的差异在于：在图15中，在n=15处，对于repeat_first_field=1, pattern=15即处在S1第15场处的“?”处，从而删除了单个场，而在图31中，由于由场“?”指定的picture_coding_type是“B-图形”，所以，repeat_first_field不被设置成1。在图33的算法中，对于n=9的场“i”，repeat_first_field变成1，在这里，时间模式(time_pattern)=5，场“i”被删除。

图32和图16之间的差异在于在图16中，在n=4处，对于repeat_first_field=1, pattern=5，即处在S1的第四场处的“C”处，从而使得单个场“?”被删除，而在图32中，由于由场“C”指定的picture_coding_type是B-图形，所以repeat_first_field不被设定成1。在图33的算法中，对于n=6的场“?”，repeat_first_field变成1，在这里，时间模式(time_pattern)等于2，场“?”

被删除。

在图31中，场“i”本来是一个产生场“I”和非间行帧的组合。在图33的算法中，由于对于B-图形不能将repeat_first_field设置成1的限制，所以场“i”被删除。其理由在于场“h”和“I”相互结合构成了一个本来就不正确的非间行帧。所以当场“J”和“j”相互结合以形成一个非间行帧时，场“i”不得被设定成单个场。这样，在当前实施例中，在根据图17的算法进行识别的过程中，由于对于B-图形不能将repeat_first_field设定成1的限制，在repeat_first_field等于1的帧(图31中 | I, h | 帧)的前后就可能产生受到扰乱而使编码效率降低的帧。通过提高与所述帧相关的位速率可以有效地克服这个缺陷。

不构成非间行帧的单个场被如上所述检测和删除。

再一次参看图4，从非间行帧检测单元178输的场信号S26被扫描转换器162按照输入序列转换成帧图形S203。可以看到，在图31和32的实施例中，S26中的“1”是所述帧边界。还可以看到在根据上述模式输出的两个场的间隔处构成了一个帧图形。

输入帧S203以及相关标记S24和S25被输入给图形编码器163。可以使用MPEG2中定义的top_first_field[n]和repeat_first_field[n]对这两个标记S24和S25进行编码。来自编码器163的编码位流S204存贮在存贮介质164中。

下面解释图4所示的译码装置。

来自图4存贮介质164的编码位流信号S205通过端176提供给图形编码器165。在图形编码器165中，从编码位流信号S205中重新构成帧信号S206。下面来解释图形判定器165。

对于帧信号S206的每一帧，用于根据top_field或bottom_field规定一个图形的第一标记S10将被显示，而用于规定所述场将被删除的第二标记S11将被译码。这些标记S10和S11与对编码器进行解释时所解释的标记S4，S5或S24，S25相关。

利用扫描转换器166将帧信号S206的每一帧转换成帧信号S207的两个场，即top_field和bottom_field。输出控制器177利用标记S10和S11控制场信号S207的图形显示。

当所述标记S10 = 1时，首先输出top_field而当S10 = 0时，则首先输出bottom_field。

若标记S11 = 1，则插入一个在前译码和再生的场，例如，根据标记S10的命令重复输出最初输出的场。在这种情况下，利用在MPEG2中定义的top_first_field和repeat_first_field，可以获得两个标记S10和S11。利用这种方式，根据信号S207和标记S10、S11输出信号S208。信号S208在监视器167上进行显示。

图35到38详细示出了输出控制器的处理。这四个例子分别示出了与结合图6至图9所解释的所述序列的编码位流相对应的输出。

下面结合图39解释图4所示图形编码器163的例子。图39所示的例子执行由公知的MPEG2中的运动补偿帧间预测和DCT相结合而组成的混合编码方法。

输入移动图形信号S203被提供给图39的端360。利用I-图形、P-图形或B-图形的三种方法中的任一种对每一帧进行编码。

在由12帧组成的GOP的情况下，如图25所示，帧I2被编码成一个I-图形，而帧B0和帧B1被编码成B-图形，帧P5被编码成一个P-图形。然后，依序列对帧B3、B4、P8、B6、B7、P11、B9和B10进

行编码。利用一个图形编码类型指定电路371来执行所述的图形编码方法，所述电路371输出信号S7。

信号S7被输入给运动矢量估计电路370，并在这里检测帧间运动矢量。通过在一个基准帧和一个当前16象素×16和宏数据块之间进行模式匹配执行运动矢量估算。即在当前宏数据块的信号 A_{ij} 和由一个运动矢量所指示的任一数据块的信号 F_{ij} 之间差值的绝对值 $|A_{ij} - F_{ij}|$ 之和 E_f ，即： $E_f = \sum |A_{ij} - F_{ij}|$ 可以获得。

运动矢量检测电路370输出运动矢量信息S43，针对于这个信息， E_f 变得最小。运动补偿帧间/帧内预测电路367输出一个利用运动矢量信息S23读出的预测图形信号S90。加法器361计算信号S203和S90之间的差值以获得预测剩余信号S91。若信号S7表示一个I-图形，则不进行预测，并直接输出所述输入运动图形信号S203。

利用具有两维DCT的DCT电路362对预测剩余信号S91进行处理。若没有进行预测，那么它就是一个利用两维DCT进行处理的原始信号，利用量化电路363对DCT系数进行纯量化，并利用变长编码(VLC)电路368对所产生的量化输出进行霍夫曼编码。在宏数据块的基础上利用VLC电路368以帧为基础对量化阶(step)Q和运动矢量信息S44进行霍夫曼编码。在帧的基础上利用VLC电路368对帧信号S7和上述标记S4和S5进行霍夫曼编码。

缓冲器369用于平滑发送通道上输出的位速率。若要出现溢出，缓冲器369工作以反馈这种影响并增加量化电路363的量化阶，同时减少从量化电路363输出的信息量。

另一方面，由量化电路363输出的数据被传送给去量化电路364，

在这里，它被去量化以响应由量化电路363提供的量化阶 Q 。去量化电路364的输出被提供给反DCT电路365，以进行反DCT处理。去量化电路364的输出被输入反DCT电路365，在这里它被进行反DCT处理。经译码的预测剩余信号S92输入给加法器366。

与提供给加法器361的预测剩余信号S90相同的一个数据被提供给加法器366。加法器366将信号S90加到信号S92上。这就产生了局部译码的图形数据，这个图形是一个与在接收机侧上的输出图形相同的信号。

参看图37到45，下面将详细解释对运动图形信号进行编码和译码的例子。在这些实施例中，输入图形信号，以用于通过插入图形信号的逐行或帧间相关性进行压缩。

如果插入了逐行相关性，可以利用例如用于数据压缩的离散余弦变换(DCT)对图形信号进一步处理。

在插入帧间相关性的情况下，可以通过编码对图形信号作进一步的压缩。例如，如果在时间点 $t = t_1, t_2, t_3$ 处产生了帧图形PC1、PC2和PC3，如图40A所示，那么就可以计算帧图形信号PC1和PC2图形信号之间的差值，以产生图形PC12，如图40B所示，同时，可以计算帧图形PC2和PC3(图40B)图形信号之间的差值以产生图形PC23，如图40B所示。由于暂时相邻帧的帧图形互相之间没有显著变化，所以，根据两个图形之间差进行计算所产生的差值信号具有很小的值。关于图40B所示的图形PC12，获得图40B中图形PC12划阴影线部分的信号，以作为图40A中帧图形PC1和PC2的图形信号之间的差值。类似的，对于图40B所示的图形PC23，获得图40B中图形PC23划阴影线部分的信号以作为图40A中帧图形PC2和PC3的图形信号之间的差

值。可以通过对该差值信号进行编码来压缩代码量。

但是，如果仅仅传送差值信号，就不能重新存贮原始图形。由此，各帧的图形被排列成三种图形，即帧内编码图形或I-图形，预测编码图形或P-图形，和双向预测编码图形或B-图形，以用于对图形信号进行的压缩编码。

即：从帧F1到帧F17帧的17帧图形信号被分组成一个图形组，以用作一个编码单元，如图41A和41B所示。导引帧的图形信号被编码成一个I-图形，而第二和第三帧F2、F3的图形信号分别被处理成一个B-图形，和一个P-图形。第四帧和随后的帧F4到F17被交替处理成B-图形和P-图形。

作为I-图形的图形信号，I-图形的一帧的图形信号被直接传送。对于P-图形的图形信号，基本上是对来自暂时在前的I-或P-图形的差值进行编码和传送，如图38A所示。对于B-图形的图形信号，基本上是获得来自暂时在前帧或暂时在后帧的平均值的差值，并对差值进行编码和传送，如图40B所示。

图42A和42B示出了用于对运动图形信号进行编码的方法的原理。特别是，图42A和42B分别示出了运动图形信号的帧数据和所传送的帧数据。在图42中，由于第一帧F1被处理成I-图形或被处理成一个非间行帧，所以，它被作为一个传送数据F1X(非间行传送数据)在传送通道上直接传送。关于第二帧，它被处理成一个B-图形或被处理成一个插入帧。获得来自暂时在前帧F1和暂时在后帧F3(帧间编码的非插入帧)之间平均值的差值，并把以产生的差值作为传送数据F2X(传送的插入帧数据)加以传送。

特别是，具有四种可在宏数据块基础上加以转换的处理成B-

图形的不同模式。第一模式是将原来帧F2的数据作为传送数据F2X直接传送，如虚线箭头SP1所示(帧间编码模式)。第二种模式是根据暂时在后帧F3计算差值，并传送所述差值，如虚线箭头SP2所示(反向预测模式)。第三种模式是传送来自暂时在前帧F1的差值，如虚线箭头SP3所示(正向预测模式)。第四种模式是产生暂时在前帧F1和暂时在后帧F3之间的差值，并将该差值作为传送数据F2X加以传送(双向预测模式)。

在这四种模式之中，在宏数据块基础上使用使传送数据最少的模式。

当传送差值数据时，来自预测图形的运动矢量X1，即正向预测帧F1和F2之间的运动矢量和/或反向预测帧F3和F2之间的运动矢量X2与差值数据一起传送。

关于P-图形的帧F3(帧间预测的非插入帧)，来自由虚线箭头SP3所指示的并作为一个预测图形的暂时在前帧F1的差值信号和运动矢量X3被计算，并被作为传送数据F3X加以传送(正向预测模式)。另外，原始帧F3的数据被作为传送数据F3X直接传送，如虚线箭头SP1所示(帧内编码模式)。在这个P-图形中，如同在B-图形中一样，在宏数据块的基础上，选择能够较大程度减少传送数据量的特定传送方法。

B-图形的帧F4和P-图形的帧F5是以类似的方式进行处理的，从而产生传送数据F4X、F5X和运动矢量X4、X5和X6。

图43示出了一个根据上述原理用于对运动图形信号进行编码，传送和译码的装置的结构。编码装置1对输入图形信号进行编码并把编码后的信号传送给作为传送通道的记录介质3，并在其上进行

记录。译码装置2再现记录在记录介质3上的信号，并译码和输出再现的信号。

在编码装置1中，输入图形信号VD通过输入端10提供给预处理电路11，在这里，它被分隔成亮度和彩色信号(色度信号)，并利用A/D转换器12和13分别对它们进行A/D转换。由A/D转换器12和13进行A/D转换成数字信号的图形信号被提供给帧存贮器14，以在其中存贮。在帧存贮器14中，亮度信号帧存贮器15和色度信号帧存贮器16分别存贮亮度和色度信号。

格式转换电路17将存贮在帧存贮器14中的帧格式信号转换成数据块格式信号。即：如图44A所示，存贮在帧存贮器14中的三种图形信号处于由其中的每一行由H个点组成的V个行组成的帧格式数据形式之中。格式转换电路17将一帧信号分成其中的每部分都由16行组成的n个部分。如图44B所示，所述的每一部分被分成多个宏数据块，每一宏数据块是由与 16×16 个象素或点对应的亮度信号构成的，如图44C所示。这些亮度信号又进一步被分成其中的每一个由 8×8 个点构成的数据块Y[1]到Y[4]。这些 16×16 点亮度信号与 8×8 点Cb信号和 8×8 点Cr信号有关。

然后，被如此转换成数据块格式的数据被从格式转换电路17提供给编码器18，在这里，如将要结合附图45在后面解释的那样，这些数据将被编码。

由编码器18编码的信号作为传送通道上的位流加以输出，从而使其记录在例如记录介质3上。

从记录介质3上再现的数据被提供给译码装置2的译码器31以进行译码。该译码器31将在后面结合图48进行解释。

由译码器31译码的数据被提供给格式转换电路32，以用于将其从数据块格式数据转换成帧格式数据。帧格式亮度信号被提供给帧存贮器33的亮度信号帧存贮器34，而色度信号被提供给色度信号帧存贮器35，以便在其中存贮它们。利用D/A转换器36、37分别对从亮度信号帧存贮器34中读出的亮度信号和从色度信号帧存贮器35中读出的色度信号进行D/A转换，随后将其提供给估处理电路38，以进行合成，所产生的输出图形信号在输出端30输出给诸如是一个CRT的显示器(未示出)，以便进行显示。

下面结合图45来解释编码器18的结构。

通过输入端49提供的用于编码的图形数据以宏数据块为基础输入给运动矢量检测电路50，然后，根据一个预定序列将帧图形数据处理成I-、P-或B-图形。陆续输入的帧图形将要被处理成按照这个预定序列的I-、P-或B-图形。例如，如图41所示，由帧F1到F17构成的图形组将被处理成I、B、P、B、P、..... B、P。

被处理成I-图形的帧，例如帧F1的图形数据被从运动矢量检测电路50传送给帧存贮器51的正向原始图形单元51a，并存贮在其中，而被处理成B-图形的帧、例如帧F2的图形数据被传送给基准原始图形单元51b，被处理成P-图形的帧、例如帧F3的图形数据被传送给反向原始图形单元51c，以便在其中存贮它们。

当在下一个时间输入被处理成B-图形的帧，例如是帧F4或被处理成P-图形的帧，例如是帧F5的图形时，至今存贮在反向原始图形单元51c中的第一P-图形帧F3的图形数据被传送给正向原始图形单元51a。另一方面，通过修正将下一个B-图形(帧F4)的图形数据存贮在基准原始图形单元51b中，而下一个P-图形(帧F5)的图

形数据通过修正被存贮在反向原始输出单元51c中。这个操作序列被重复执行。

读出存贮在帧存贮器51中各图形的信号，并利用预测模式转换电路52以帧预测模式和场预测模式进行处理。在预测判定电路54的控制之下，处理单元53还利用帧内模式，正向预测模式，反向预测模式或双向预测模式对各图形信号进行处理。使用这些预测模式中的哪一个是根据在宏数据块基础上的预测误差信号确定的，即根据基准信号和预测图形之间的差值来确定的。这样，运动矢量检测电路50产生用于这种判定的预测误差信号绝对值的和以及在宏数据块的基础上与预测误差信号相应的帧内编码模式鉴定值。可以使用平方和来取代绝对值的和。

下面解释预测模式转换电路52中的帧预测模式和场预测模式。

若设定了帧预测模式，预测模式转换电路52把由运动矢量检测电路50提供的四个亮度数据块Y[1]到Y[4]直接输出给下游侧处理单元53。在这种情况下，偶数场的行数据与奇数场的行数据共存，如图46A所示。在图46A和46B中，在每个宏数据块内的实线和虚线分别表示奇数场行(第一场的行)数据和偶数场的行(第二场的行)数据，而a和b(图44a和图44b)规定运动补偿的单元。在帧预测模式下，预测是根据4个作为一个单元的亮度数据块(宏数据块)进行的，并且每一个运动矢量都涉及到4个亮度数据块。

若设定了场预测模式，预测模式转换电路52将使从具有图46A所示结构的运动矢量检测电路50输入的4个亮度数据块中的亮度数据块Y[1]和Y[2]仅由例如奇数场行的点构成，而使其余的两个亮度数据块Y[3]和Y[4]仅由偶数场的行数据构成。所产生的数据被输出

给处理单元53。在这种情况下，一个运动矢量和两个亮度数据块Y[1]和Y[2]相关，而另一个运动矢量和其余的两个亮度数据块Y[3]和Y[4]相关。

在帧预测模式下，色度信号在奇数场的行数据和偶数场的行数据共存的状态下被提供给处理单元53，如图46A所示。在场预测模式下，色度数据块Cb和Cr的上半部分(4行)是与亮度数据块Y[1]和Y[2]对应的奇数场的色度信号，而色度数据块Cb和Cr的下半部分(4行)是与亮度数据块Y[3]和Y[4]对应的偶数场的色度信号，如图46B所示。

运动矢量检测电路50在预测判定电路54中产生帧内编码模式的鉴定值和预测误差绝对值的和，以确定帧内编码模式，正向预测模式、反向预测模式或双向预测模式中的哪一个和帧预测模式或场预测模式中的哪一个被逐个从一个到另一个地用于各宏数据块。

就是说，要获得在需要编码的基准图形宏数据块的信号 A_{ij} 和它的平均值之间差值的绝对值之和、即 $\sum |A_{ij} - (A_{ij} \text{的平均值})|$ 以同作帧内编码模式的鉴定值。获得在基准图形宏数据块信号 A_{ij} 和预测图形 $(A_{ij} - B_{ij})$ 宏数据块信号 B_{ij} 之间差值的绝对值之和，即： $\sum |A_{ij} - B_{ij}|$ 作为正向预测的预测误差的绝对值之和。在预测图形被改变成与正向预测图形不同的情况下，以类似的方式获得反向预测和双向预测的预测误差绝对值的和以用于帧预测模式和场预测模式的每一种。

这些绝对值的和被提供给预测判定电路54，该电路54从与帧预测模式和场预测模式相关的正向预测，反向预测和双向预测的预测误差绝对值之和中选择最小的一个作为与帧间预测相关的预测误差

绝对值的和。帧间预测误差绝对值的和也和帧内编码模式的鉴定值进行比较，并选择其中较小的一个。与如此选择的值相关的模式被选择预测模式和帧/场预测模式。就是说，如果帧内编码模式的鉴定值较小，就选择帧内编码模式。如果帧间预测误差绝对值的和较小，那么，正向预测模式、反向预测模式或双向预测模式中具有最小绝对值之和的一个被选作预测模式和帧/场预测模式。

如上所述，预测模式转换电路52向图46结构中的处理单元53提供与由预测判定电路54所选择的帧预测模式或场预测模式相对应的基准图形的宏数据块信号。运动矢量检测电路50输出在预测图形和基准图形之间的与由预测判定电路54所选预测模式相应的运动矢量，并将该运动矢量提供给变长编码电路58和运动补偿电路64，这在下面还要解释。关于这个运动矢量，要选择与预测误差绝对值最小和相关的运动矢量。

当运动矢量检测电路50从正向原始图形单元51a中读出I-图形的图形数据时，预测判定电路54设定帧内编码模式(不执行运动补偿的模式)并将处理单元53的开关53d置于触点a，这就把I-图形的图形数据传送给了DCT模式的转换电路55。

DCT模式转换电路55将四个亮度数据块的数据设置成奇数场的行和偶数场的行共存的状态(帧DCT模式)或奇数场的行和偶数场和行彼此隔离的状态(场DCT模式)，如图47A和47B所示。然后，DCT模式转换电路55把数据输出给DCT电路56。

DCT模式转换电路55把对由奇数场数据和偶数场数据共同构成的数据执行DCT处理所实现的编码系数和对由彼此相互隔开的单数场数据和偶数场数据组成的数据执行DCT处理所实现的编码系数进

行比较，并选择具有较高编码系数的模式。

输入信号被设置成如图47A所示的奇数场的行和偶数场的行共存的状态。计算在奇数场和偶数场垂直相邻行信号之间的差，并获得绝对值的和或平方和的和。另外，输入信号被设置成为如图4B所示的奇数场的行和偶数场的行彼此相互绝缘的状态，计算奇数场和偶数场垂直相邻行信号之间的差，并获得绝对值或平方和的和。这两个绝对值的和相互比较，并设定与较小值对应DCT模式。假如前者较小，则设定帧DCT模式，反之，则设定场DCT模式。

与所选DCT模式对应的结构数据被输出给DCT电路，而用以规定所选择DCT模式的DCT标记被输出给变长编码电路58。

通过对在预测模式转换电路52(图46)中的帧/场预测模式和在DCT转换电路55(图47)中的DCT模式进行比较可以看出，在两种模式下的亮度数据块的数据结构基本保持相同。

如果在预测模式转换电路52中选择了帧预测模式(偶数行和奇数行共存的模式)，那么，在DCT模式转换电路55中选择帧DCT模式(偶数行和奇数行共存的模式)的概率是很高的。另一方面，如果在预测模式转换电路52中选择了场预测模式(偶数行和奇数行彼此相互绝缘的模式)，那么，在DCT模式转换电路55中类似选择场DCT模式(偶数行和奇数行彼此相互绝缘的模式)的概率是很高的。

但这种情况并不是必然的，预测模式转换电路52应当设定成那种预测误差绝对值的和最小的模式，而DCT模式转换电路55要设定成使编码系数最佳化的模式。

由DCT模式转换电路55输出的I - 图形数据被提供给DCT电路56，借此进行DCT处理，以将其转换成DCT系数。DCT系数被提供给量化

电路57, 并在被提供给VLC电路58之前利用与在传送缓冲器59中的数据存贮量相应的量化步进行量化。

VLC电路58把根据由量化电路57提供的量化步(比例)进行量化并从量化电路57输出的图形数据(I-图形数据)转换成诸如霍夫曼代码的变长码, 然后输出给传送缓冲器59。

VLC电路58还被提供有分别来自量化电路57, 预测判定电路54, 运动矢量检测电路50, 预测判定电路54和DCT模式转换电路55量化步(比例)、预测模式(指示帧内编码模式, 正向预测模式, 反向预测模式或双向预测模式中哪一种已被设定的一种模式)、运动矢量、预测标记(规定帧预测模式或场预测模式中哪一个已被设定的标记)和DCT标记(规定帧预测模式或场预测模式中的哪一个已被设定的标记), 以用于变长编码。

传送缓冲器59暂存输入的数据, 并将与存贮数据量相应的数据输出给量化电路57。

如果传送缓冲器59中的剩余数据量增加到上容许极限值, 那么, 通过量化控制信号使传送缓冲器增加量化比例, 以减少被量化数据的数据量。相反, 若传送缓冲器59中剩余的数据量减少到下容许极限值, 则利用量化控制信号使传送缓冲器去减少量化比例, 以增加被量化数据的数据量。这就防止了传送缓冲器59产生上溢或下溢。

在一个预定时间处从传送缓冲器59中读出数据, 并通过输出端69在传送通道上加以输出, 以便存贮在例如记录介质3上。

由量化电路57输出的I-图形数据被提供给去量化电路60, 并在这里根据由量化电路57提供的量化步进行去量化。去量化电路60的输出被提供给反DCT(IDCT)电路61, 并进行反DCT处理, 随后通过

加法器62被提供给帧存贮器63的正向预测图形单元63a, 以便在其中进行存贮。

当将陆续输入的各帧图形数据处理成I、B、P、B、P、B.....图形时, 运动矢量检测电路50将最初输入的帧图形处理成I-图形, 并且在把下一帧的图形数据处理成B-图形之前, 把紧跟在下一帧之后的一个帧的图形数据处理成P-图形。由于B-图形很可能被伴随有反向预测和正向预测, 所以, 如果没有预先提供作为反向预测图形的P-图形, 它是不能被解码的。

随后, 运动矢量检测电路50开始对存贮在反向原始图形单元51c内的P-图形的图形数据进行处理。和在上述情况中一样, 以帧内编码模式的鉴定值为基础的宏数据块和以帧间差值(预测误差)为基础的宏数据块被从运动矢量检测电路50提供给预测判定电路54。预测判定电路54响应在宏数据块基础上的预测误差的绝对值的和以及帧内编码数据块的鉴定值, 从一个宏数据块到另一块地设定帧预测模式或场预测模式的一个模式和帧内编码模式或正向预测模式这一个模式。

当设定了帧内编码模式时, 如上所述, 处理单元53将开关53d置于触点a。与I-图象数据类似, 这些数据被通过DCT转换电路55、DCT电路56、量化电路57、VLC电路58和传送缓冲器59传送给传送通路。这些数据还通过去量化电路60、IDCT电路61和加法器62提供给帧存贮器63的反向预测图形单元63b, 以在其中存贮。

对于正向预测模式, 开关53d被置于触点上, 同时, 从帧存贮器63的正向预测图形单元63a中读出所存贮的图形数据, 这里是I-图形数据, 并利用运动补偿电路64根据由运动矢量检测电路50提供

的运动矢量对这些数据进行运动补偿。就是说，如果预测判定电路54命令设定正向预测模式，那么运动补偿电路64利用对应于从与由运动矢量检测电路50正在输出的宏数据块的位置相对应的位置开始的运动矢量的移位来读出正向预测图形单元63a的读出地址，以产生预测图形数据。

由运动补偿电路64输出的预测图形数据被馈送给加法器53a。加法器53a从由预测模式转换电路52提供的基准图形的宏数据块数据中减去与所述宏数据对应的来自运动补偿电路64的预测图形数据，并输出所获得的差值。这个差值数据通过DCT模式转换电路55、DCT电路56、量化电路57、VLC电路58和传送缓冲器59传送给传送通路。这个差值还被IDCT电路61局部译码并提供给加法器62。

加法器62也被提供有与提供给加法器53a相同的预测图形数据。加法器62把来自运动补偿电路64的预测图形数据加到由IDCT电路61输出的差值数据上。这就给出了原始译码的P-图形数据。P-图形的图形数据被提供给帧存贮器63的反向预测图形单元63b，以在其中存贮。由于事实上由IDCT电路输出的差值数据的数据结构必须与预测图形数据的数据结构相同，所以，在帧/场预测模式和帧/场DCT模式不同的情况下，必须是提供一个电路，以重新排列这些数据。但为简单起见，这个电路被省略掉了。

在将I-图形和P-图形的数据以这种方式分别存贮在正向预测图形单元63a和反向预测图形单元63b之中以后，运动矢量检测电路50执行有关B-图形的处理。预测判定电路54设定所述的帧/场预测模式以响应帧间差值绝对值的和以及以帧内编码模式为基础的宏数据块的鉴定值。预测判定电路54还把预测模式设定成帧内编码、正

向预测、反向预测和双向预测模式的一个模式。

对于帧内编码或正向预测模式，开关53d被置于触点a或b。所作处理类似于对P-图形所作的有关数据传输的处理。

对于反向预测模式，开关53d被置于触点e，存贮在反向预测图形单元63b中的图形数据，这里是P-图形的图形数据被读出，并利用运动补偿电路64根据从运动矢量检测电路50输出的运动矢量对这些数据进行运动补偿。即：当预测判定电路54命令设定反向预测模式时，运动补偿电路64使反向预测图形单元63b的读出地址从与正在被运动矢量检测电路50输出的宏数据块的位置相应的位置开始移动一个相当于运动矢量的量，以产生预测图形数据。

由运动补偿电路64输出的预测图形数据被馈送给加法器53b。加法器53b从由预测模式转换电路52提供的基准图形的宏数据块数据中减去来自运动补偿电路64并与所述宏数据块相对应的预测图形数据，并输出所产生的差值。该差值数据通过DCT模式转换电路55、DCT电路56、量化电路57、VLC电路58和传送缓冲器59传送给传送通道。

对于双向预测模式，开关53d被置于触点d，读出存贮在正向预测图形单元63a中的图形数据，这里是I-图形数据并读出存贮在反向预测图形单元63b中的图形数据，这里是P-图形数据，并利用运动补偿电路64根据由运动矢量检测电路50输出的运动矢量对这些数据进行运动补偿。即：当运动判定电路54命令设定反向预测模式时，运动补偿电路64使反向预测图形单元63b的读出地址从与由运动矢量检测电路50正在输出的宏数据块的位置相对应的位置开始移动一个与运动矢量相应的量，以产生预测图形数据。

由运动补偿电路64输出的预测图形数据被馈送给加法器53c。加法器53c从由运动矢量检测电路50提供的基准图形的宏数据块中减去来自运动补偿电路64并与所述宏数据块相应的预测图形数据并输出所产生的结果。这个差值数据通过DCT模式转换电路55、DCT电路56、量化电路57、VLC电路58和传送缓冲器59传送给传送通路。

由于B-图形的图形不被用作其它图形的基准图形，所以，它不被存贮在帧存贮器63中。

在帧存贮器63中，无论在何时需要，正向预测图形单元63a和反向预测图形单元63b都是被成组转换的，所以，存贮在这些单元之一中的数据可以被有选择地输出，以用作正向预测图形成反向预测图形，从而给出基准图形。

虽然前面的叙述只涉及到了亮度数据块，但色度信号也作为一个单元根据图46和47所示的宏数据块进行类似的处理。对于与色度数据块处理相关的运动矢量，使用了与相关亮度数据块相关的、在垂直和水平方面上幅值减半的运动矢量。

参看图48，它以方框图的形式示出了译码器31一个实施例的结构。一个未示出的接收电路接收通过传送通路(记录介质3)传送的图形数据或利用一个重新构成单元重新构成这些图形数据。然后，图形数据经过输入端81被暂存在接收缓冲器81中，而后被提供给译码电路90的VLC电路82。VLC电路82对由接收缓冲器81提供的数据进行变长编码，并向运动补偿电路87输出运动矢量、预测模式、预测标记和DCT标记，同时间去量化电路83输出量化步和译码的图形数据。

去量化电路83根据由VLC电路82提供的量化步对由VLC电路82提

供的图形数据进行去量化处理，并向IDCT电路84输出去量化的数据。利用IDCT电路84对去量化电路83输出的数据(DCT系数)进行ICDT处理，然后提供给加法器85。

假如由IDCT电路84提供的图形数据是I-图形数据，则加法器85输出该数据，并提供给和存贮在帧56的正向预测图形单元86a中，用以产生用于陆续输入给加法器85的图形数据(P-或B-图形数据)的预测图形数据。所述数据还在格式转换电路32(图43)处被输出。

如果由IDCT84提供的图形数据是具有作为预测图形数据的直接在步帧的图形数据的P-图形数据和利用正向预测模式编码的宏数据块数据，那么，存贮在帧存贮器86的正向预测单元86a中的直接在步帧的图形数据被读出，并利用运动补偿电路87根据从VLC电路82输出的运动矢量对其作运动补偿。加法器85把经过运动补偿的数据加到由IDCT电路84提供的图形数据(差值数据)上。所产生的数据，即译码后的P-图形数据被提供给帧存贮器86的反向预测图形单元86b，以用于产生随后提供加法器85的图形数据(B-图形成P-图形数据)的预测图形数据。

与I-图形的数据相似，加法器85不对以帧内编码模式进行编码的宏数据块数据进行规定处理，并将其直接存贮在反向预测图形单元86b中，尽管所述数据是P-图形数据。

如果由IDCT电路84提供的图形数据是B-图形数据，那么，读出存贮在帧存贮器86的正向预测图形单元86a中的I-图形的图形数据(对于正向预测模式情况)和/或存贮在帧存贮器86的反向预测图形单元86b中的P-图形的图形数据(对于反向预测模式情况)并利用运动补偿电路87根据从VLC电路82输出为运动矢量对这些数据进行

运动补偿，以产生预测图形。若不需要进行运动补偿，如同在帧内编码的情况下，那么就不会产生预测图形。

加法器85将运动补偿电路87输出的经过运动补偿的数据加到IDCT电路84的输出上。相加以后的输出通过输出端91输出给格式转换电路32。

但是，这个和输出是一个B-图形数据，并不被用于产生用于其它图形的预测图形，所以，它不被存贮在帧存贮器86中。

在输出了B-图形之后，读出存贮在反向预测图形单元86b的P-图形的图形数据，并作为重放图形通过运动补偿电路87和加法器85加以输出。但是在这时不执行运动补偿和加法。

虽然在译码器31中没有示出与图45编码器18中的预测模式转换电路52和DCT模式转换电路55相应的电路，但是利用运动补偿电路87执行了由这些电路所执行的处理，即：用于把奇数场和偶数场的行信号的分离结构重新存贮到奇数场和偶数场和行信号共存的原来结构中的处理。

虽然在前面仅描述了有关亮度信号的处理，但色度信号也是以类似方式进行处理。色度信号所使用的运动矢量长度在垂直和水平方向上都是亮度信号所使用的长度的一半。

来自预处理电路104的三种标记，即：picture_coding_type、top_field_first和repeat_first_field与图形数据一起被提供给编码装置1，以控制作为编码单元的帧或场组，来根据与原始非间行图形相对应的编码帧实现编码。另外，还对I-、P-和B-图形的类型加以控制，以避免把上述的场重复编码成B-图形。

本发明并不局限于上述实施例。例如，可以使用各种不同结构

的不规则模式检测电路或非间行图形重新构成电路。若在第二场中产生上述单个场，包括第二场重复的场组可以被重新构成并进行传输以进行编码。本发现还可以应用在不插入记录介质的情况下传送信号。

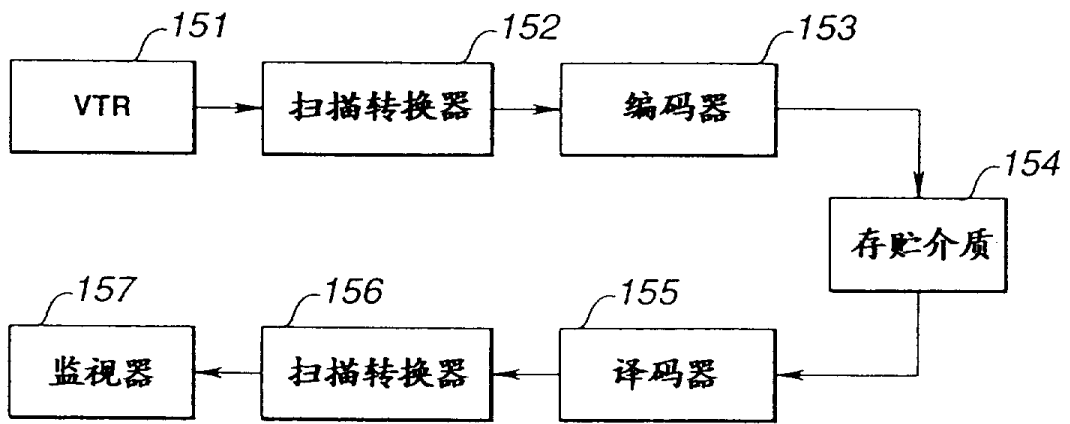


图 1

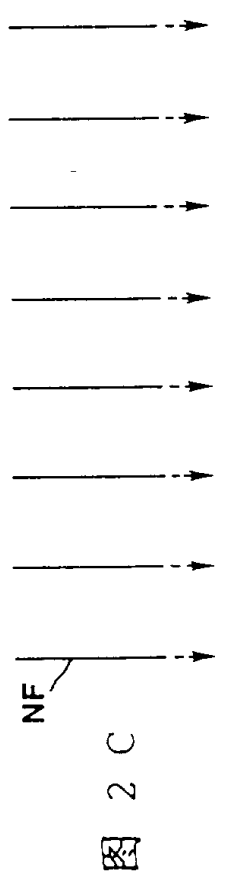


图 2 A

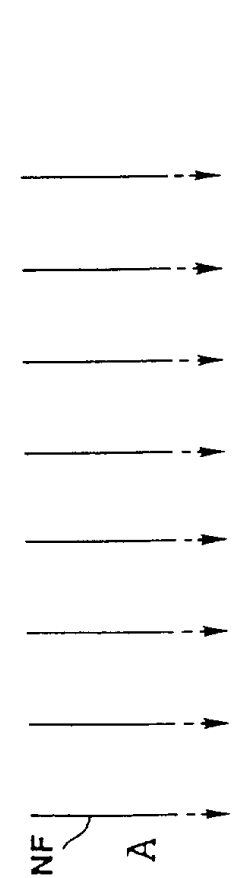


图 2 B

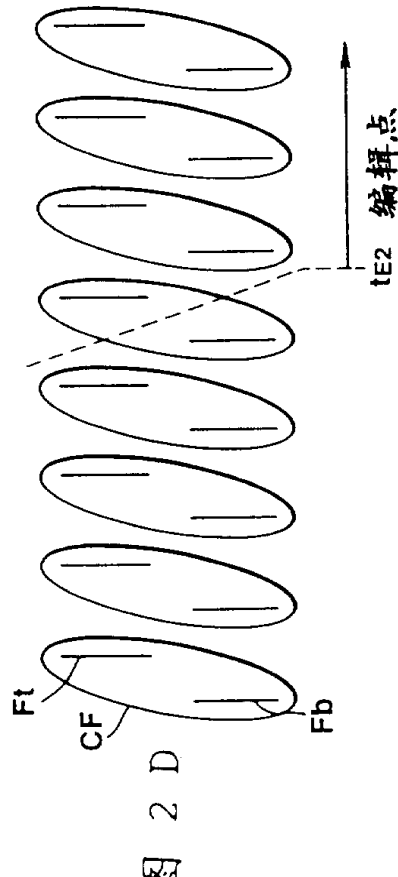


图 2 C

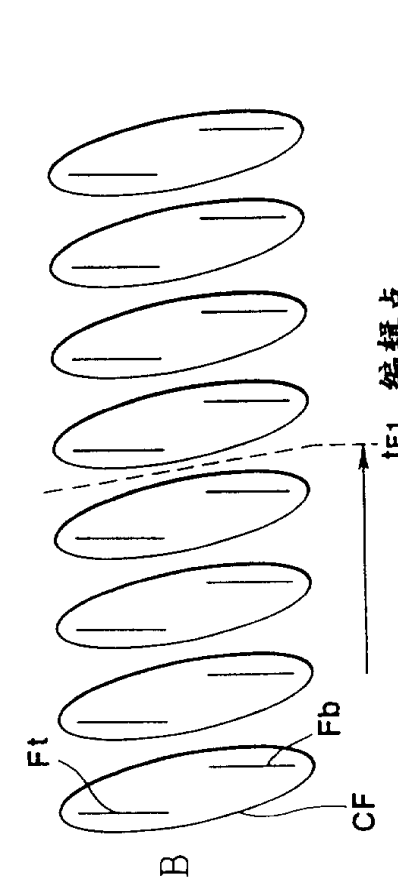


图 2 D

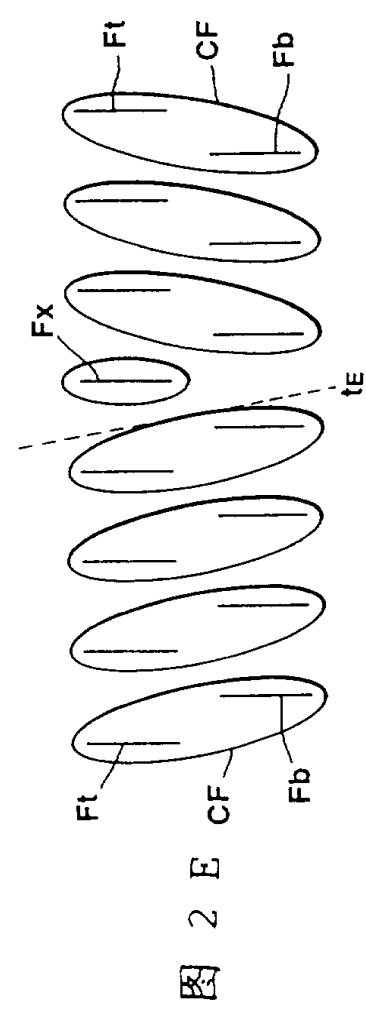


图 2 E

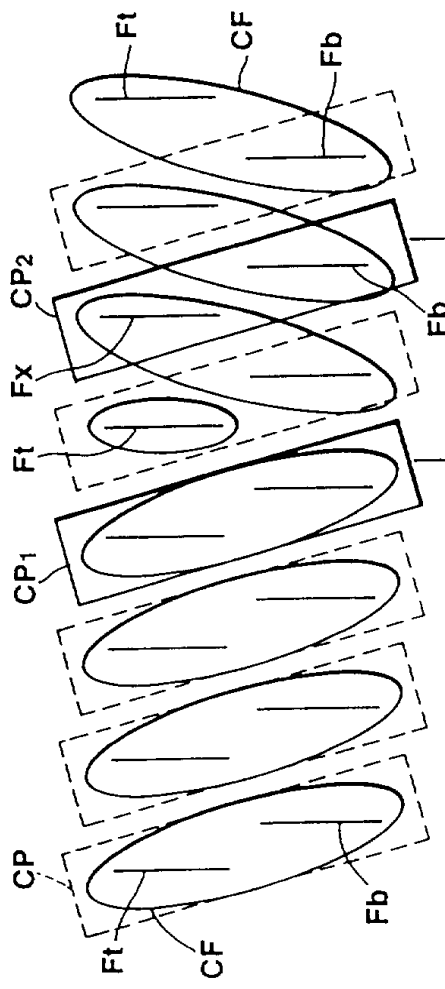


图 3 A

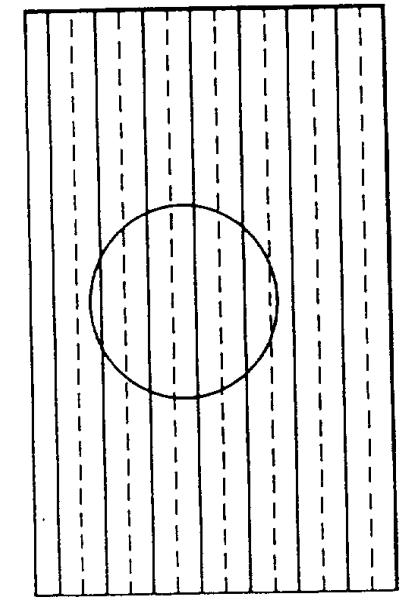


图 3 B

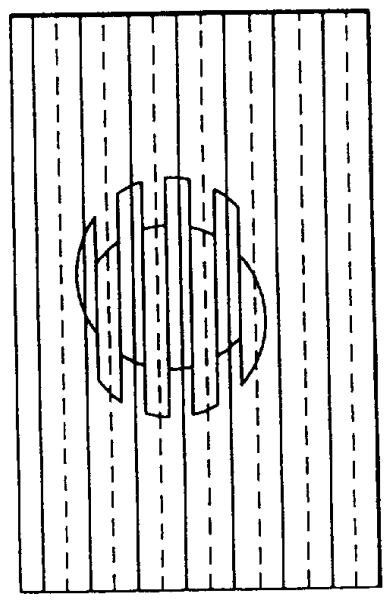


图 3 C

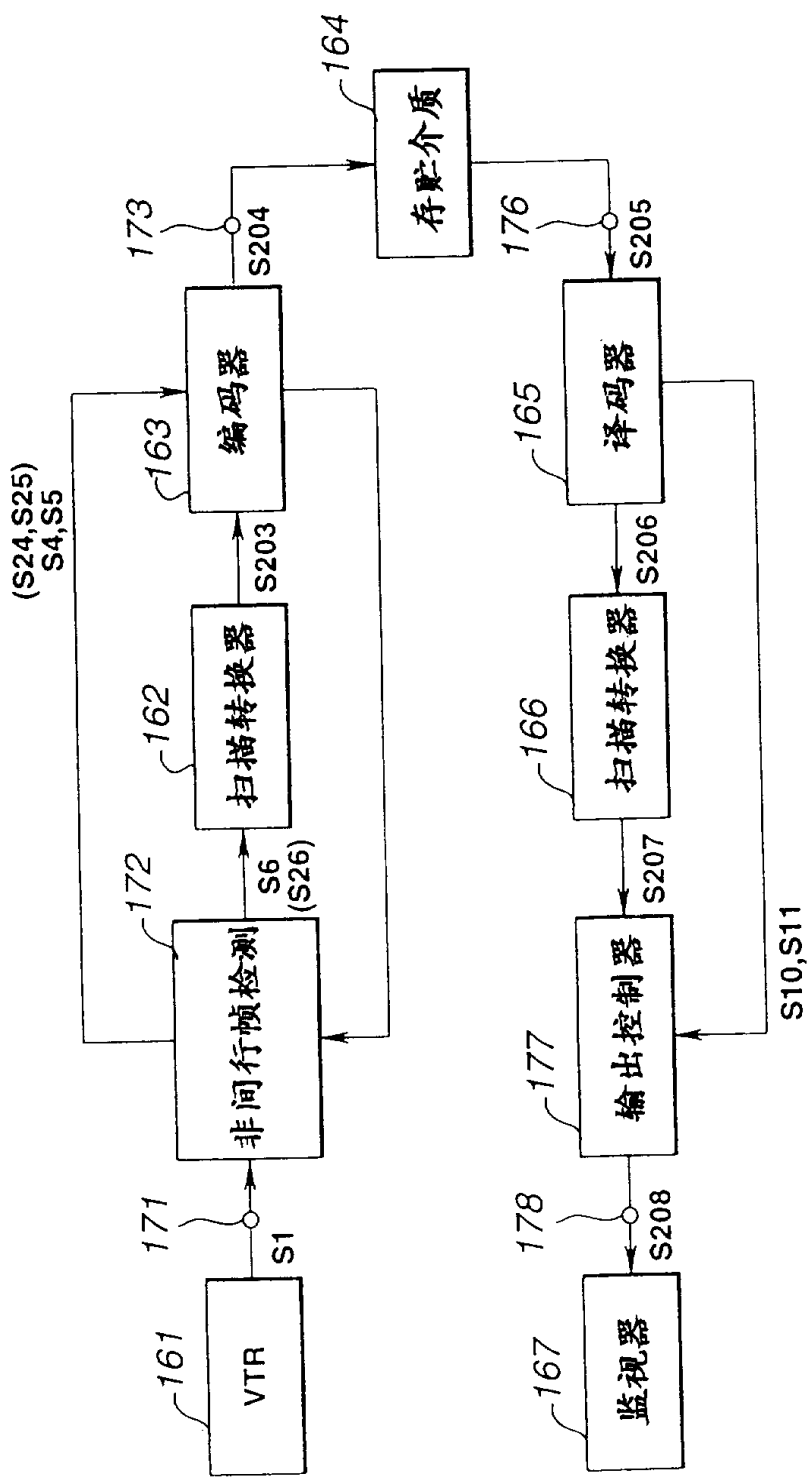


图 4

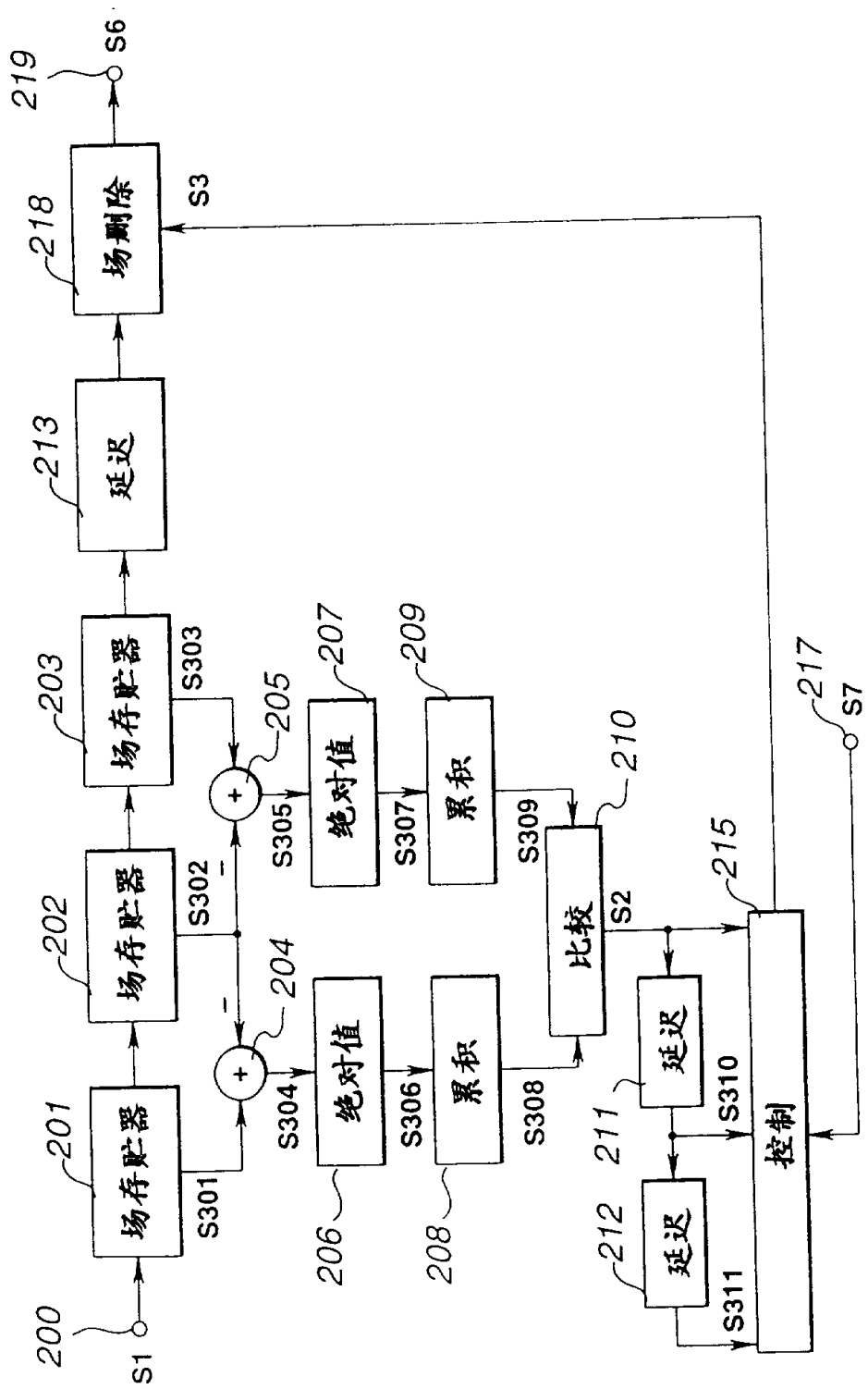


图 5

编辑点



S1	A	a	B	b	C	c	?	d	D	e	E	f	F	g
S2	1	0	1	0	1	0	1	1	0	1	0	1	0	1
S3	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S4	1		1		1		1		0		0		0	
S5	0		0		0		1		0		0		0	
S6	A	a	B	b	C	c	d	D	e	E	f	F		

图 6

编辑点



S1	A	a	B	b	C	c	?	d	D	e	E	f	F	g
S2	1	0	1	0	1	0	0	1	0	1	0	1	0	1
S3	0	0	0	0	0	0	1	0	0	0	0	0	0	0
S4	1		1		1			0		0		0		
S5	0		0		1			0		0		0		
S6	A	a	B	b	C	c	d	D	e	E	f	F		

图 7

编辑点



S1	D	e	E	f	F	g	G	?	H	h	I	i	J	j	K	k
S2	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
S3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
S4		0		0		0		0		1		1		1		1
S5		0		0		0		1		0		0		0		0
S6		e		f		g				h		i		j		k

图 8

编辑点



S1	D	e	E	f	F	g	G	?	H	h	I	i	J	j	K	k
S2	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0
S3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
S4		0		0		0				1		1		1		1
S5		0		0		0				0		0		0		0
S6		e		f		g				h		i		j		k

图 9

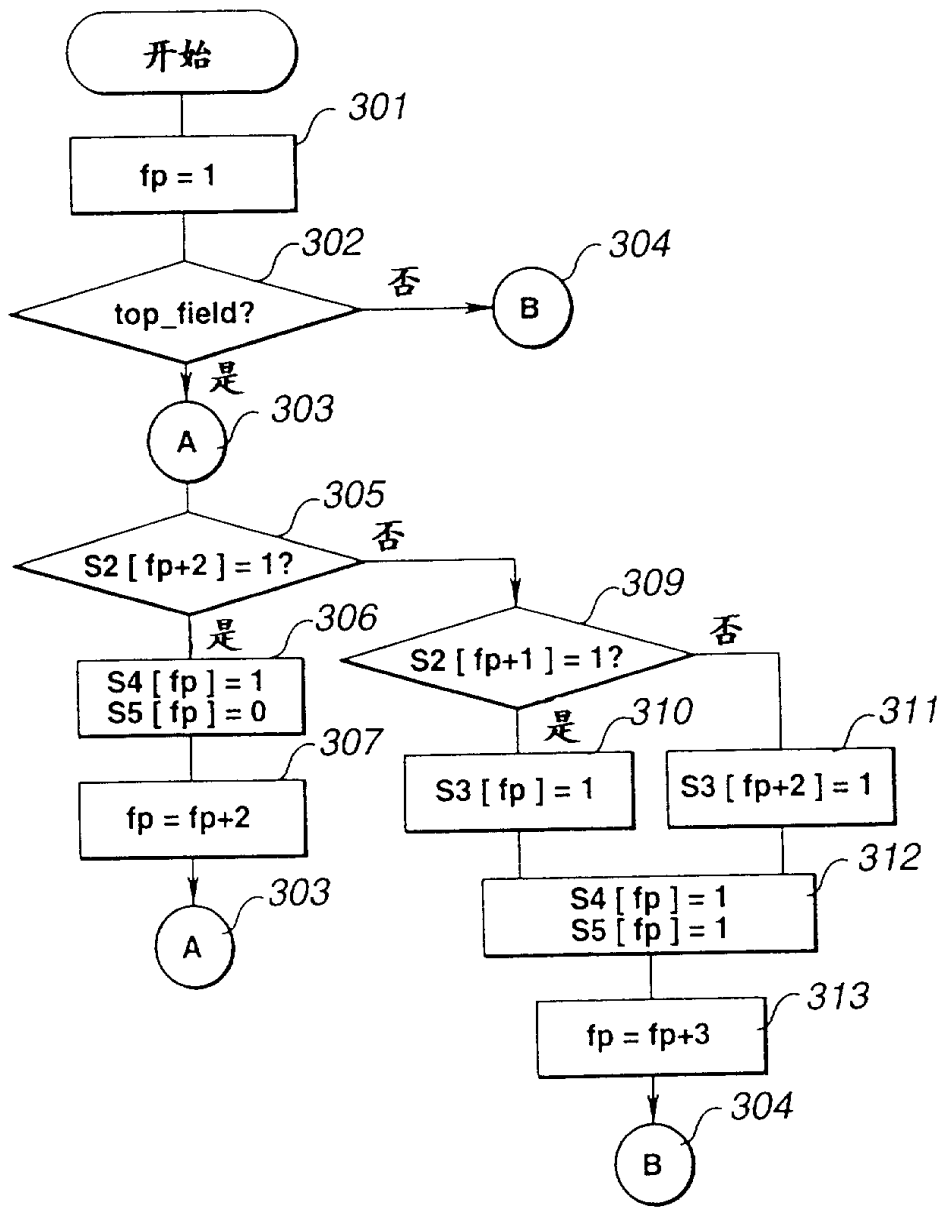


图 10

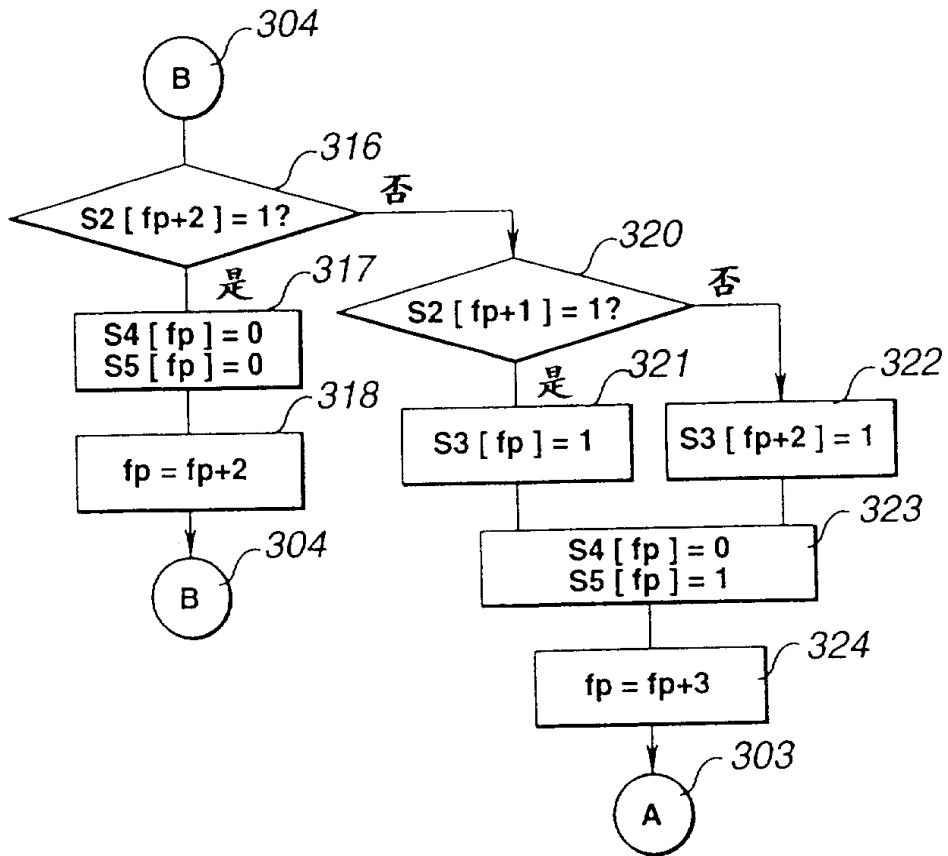


图 11

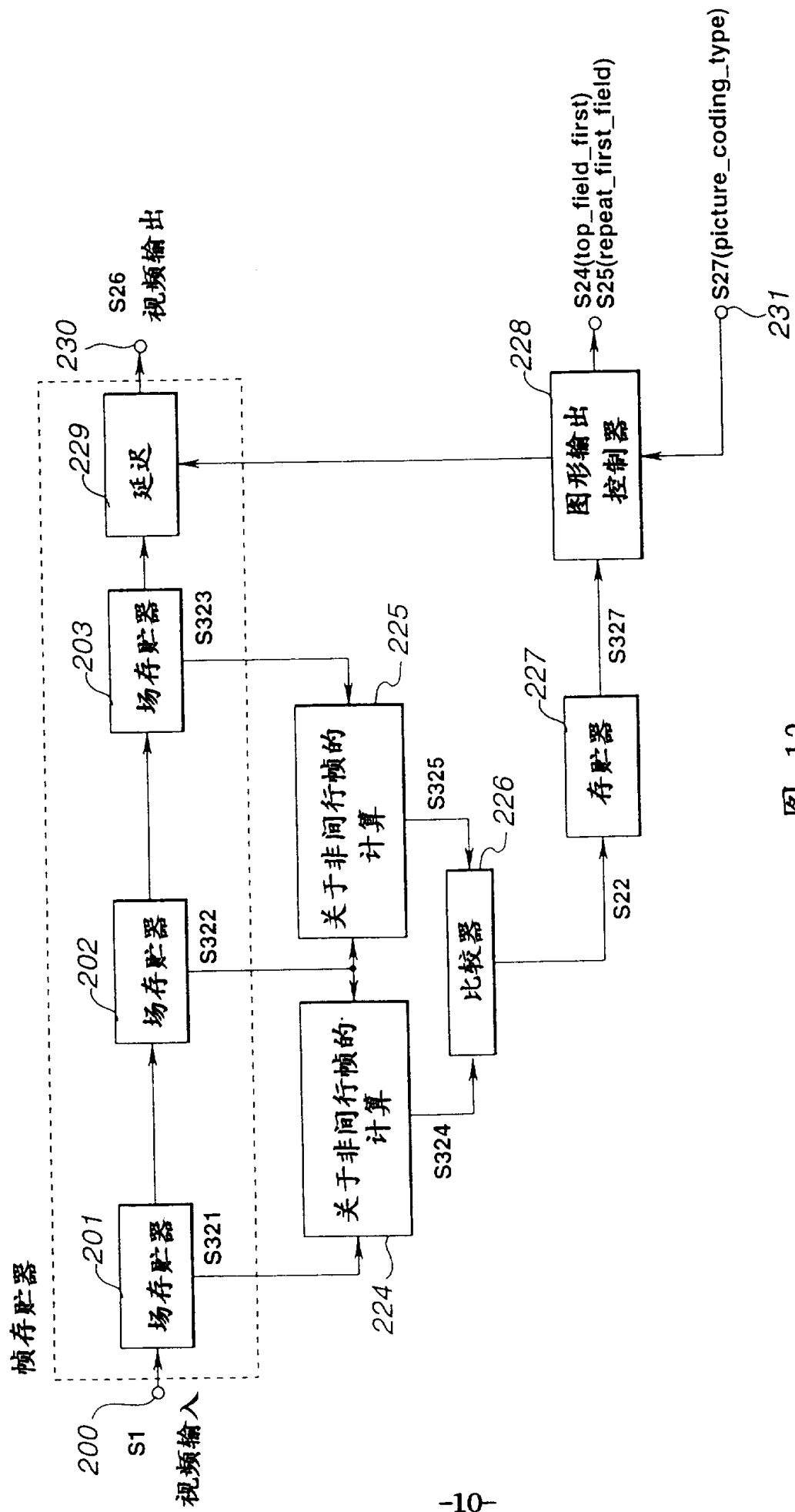


图 12

```

1  var = 0;
2  for(pix=0; pix<frame_horizontal_size; pix++) {
3      for(line=0; line<frame_vertical_size-2; line+=2) {
4          dif1 = F(pix, line ) - F(pix, line+1);    ... (a)
5          dif2 = F(pix, line+1) - F(pix, line+2);    ... (b)
6
7          var += |dif1| + |dif2| ;                  ... (c)
8      }
9  }

```

图 13

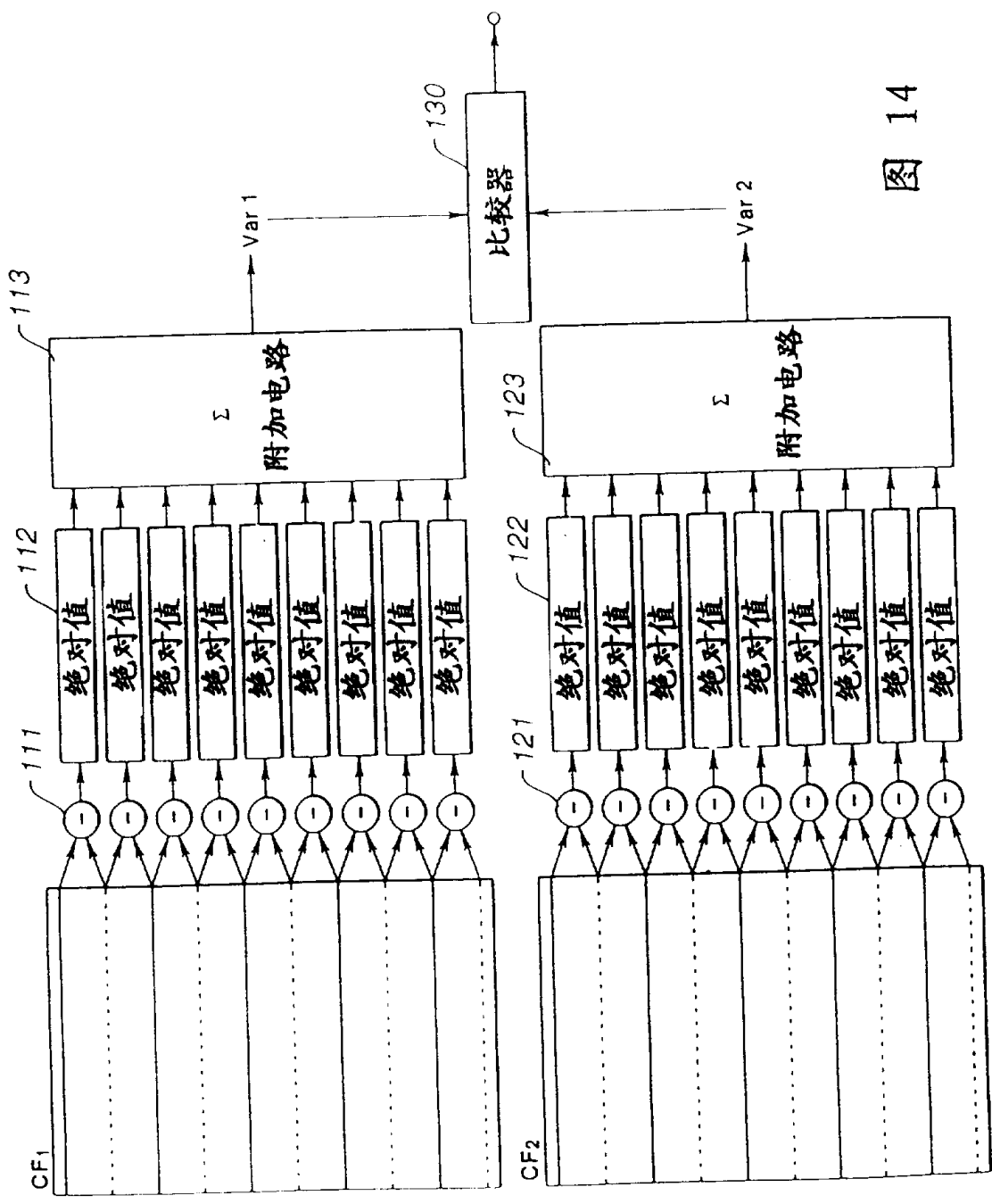


图 14

编辑点

编辑点



S1	A	B	C	?	D	E	F	G	H	I	J	K
	a	b	c	d	e	f	g	?	h	i	j	k
S22	1	0	1	0	1	0	1	0	1	0	1	0
S24(tff)	1	1	1	0	0	0	0	0	1	1	1	1
S25(rff)	0	0	1	0	0	0	1	0	0	0	0	0
模式	1	1	2	4	4	4	5	1	1	1	1	1
S103	A	B	C	?	D	E	F	G	H	I	J	
	a	b	c	d	e	f	g	?	h	i	j	
S104	A	B	C	?	D	E	F	G	H	I	J	
	a	b	c	d	e	f	g	?	h	i	j	
S105	A	B	C	?	D	E	F	G	H	I	J	
	a	b	c	d	e	f	g	?	h	i	j	
S26	A	B	C	D	-	E	F	G	H	I	J	
	a	b	c	d	-	e	f	g	h	i	j	
帧延迟	4	4	4	4	3	4	-	4	5	4	4	4
	4	4	4	4	4	4	5	4	4	4	4	4

1/3

图 15

编辑点

编辑点

S1	A	B	C	?	D	E	F	G	H	I	J	K
	a	b	c	d	e	f	g	?	h	i	j	k
S22	1	0	1	0	0	1	0	1	0	0	1	0
S24(tff)	1	1	1	0	0	0	1	1	1	1	1	1
S25(tff)	0	1	1	0	0	1	0	0	0	0	0	0
模式	1	1	3	4	4	4	6	1	1	1	1	1
S103	A	B	C	?	D	E	F	G	H	I	J	
	a	b	c	d	e	f	g	?	h	i	j	
S104	A	B	C	?	D	E	F	G	H	I	J	
	a	b	c	d	e	f	g	?	h	i	j	
S105	A	B	C	?	D	E	F	G	H	I	J	
	a	b	c	d	e	f	g	?	h	i	j	
S26	A	B	C	-	D	E	F	G	H	I	J	
	a	b	c	-	d	e	f	g	h	i	j	
帧延迟	4	4	4	4	4	5	4	5	4	4	4	

图 16

```

1  function ( ) {
2      n = 0;
3      isTop_field = 1;
4
5      while (!EOF) {
6
7          if (S22[n+3] == 1 &&
8              S22[n+4] == 0 &&
9              S22[n+5] == 1 &&
10             S22[n+6] == 0) {
11
12             if (isTop_field == 1) {
13                 if (S22[n+1] == 1)
14                     top_field_first[n] = 1, repeat_first_field[n] = 1, pattern[n] = 2;
15                 else
16                     top_field_first[n] = 1, repeat_first_field[n] = 1, pattern[n] = 3;
17
18                 isTop_field = 0;
19             }
20             else {
21                 if (S22[n+1] == 1)
22                     top_field_first[n] = 0, repeat_first_field[n] = 1, pattern[n] = 5;
23                 else
24                     top_field_first[n] = 0, repeat_first_field[n] = 1, pattern[n] = 6;
25
26                 isTop_field = 1;
27             }
28
29             n = n+3;
30
31         }
32         else {
33
34             if (isTop_field == 1)
35                 top_field_first[n] = 1, repeat_first_field[n] = 0, pattern[n] = 1;
36             else
37                 top_field_first[n] = 0, repeat_first_field[n] = 0, pattern[n] = 4;
38
39             n = n+2;
40
41         }
42     }
43
44 }

```

图 17

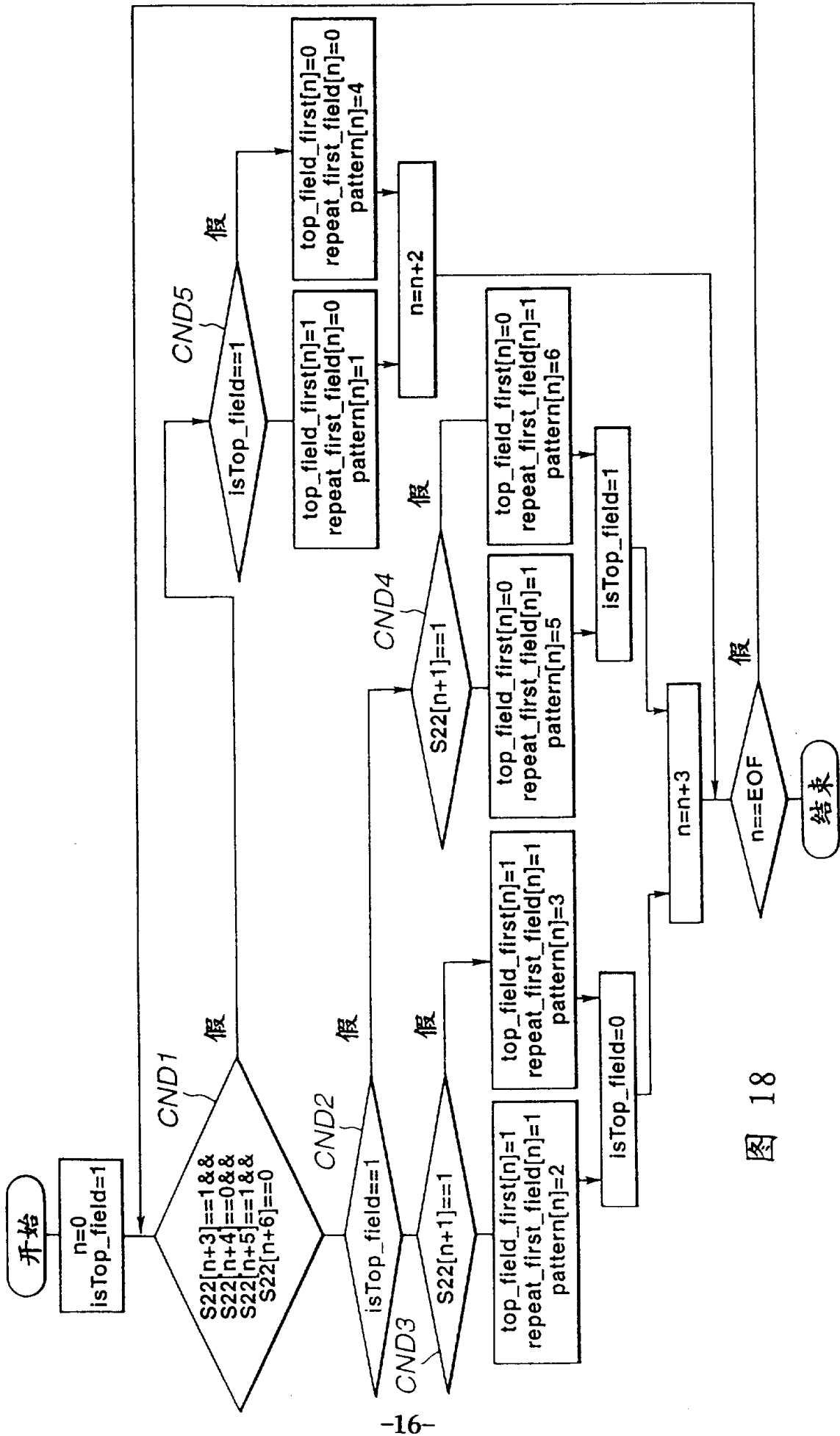


图 18

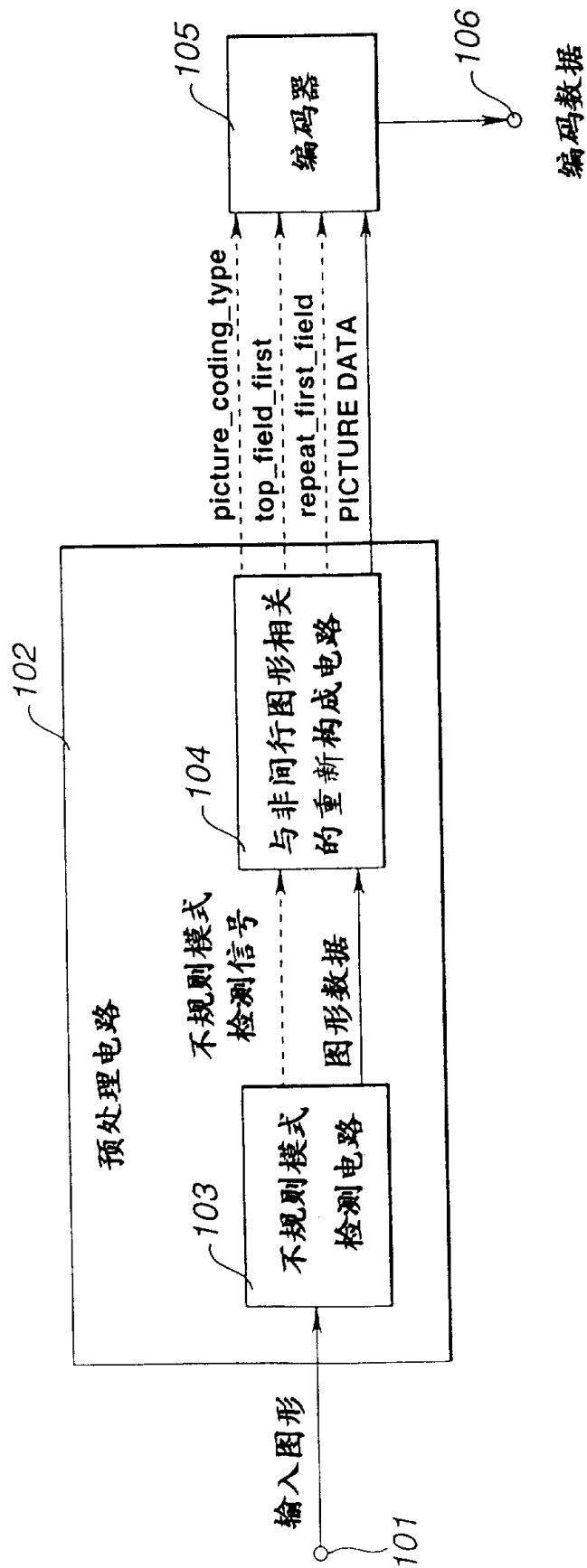


图 19

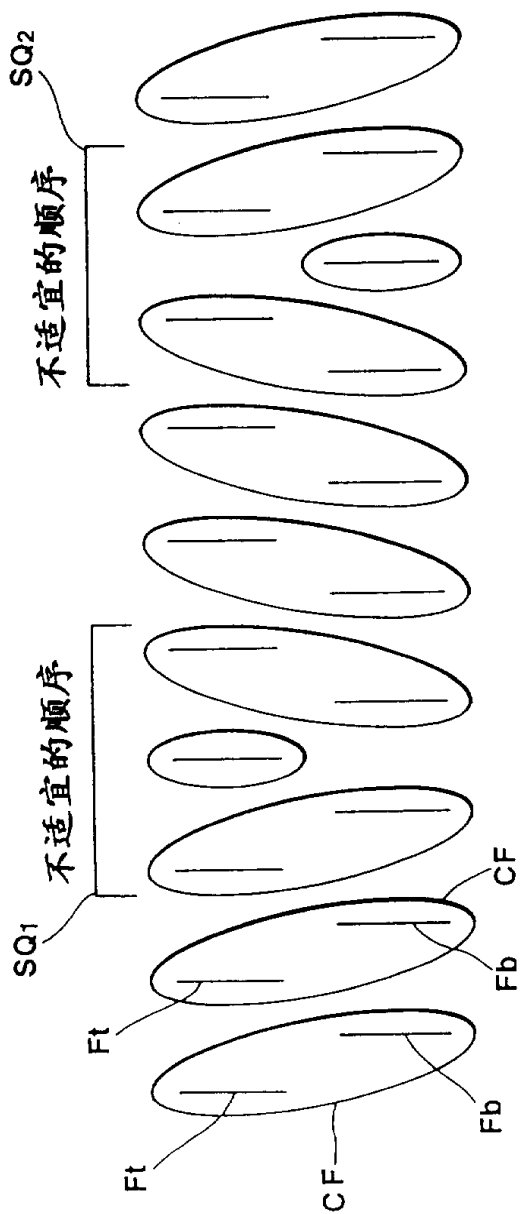


图 20

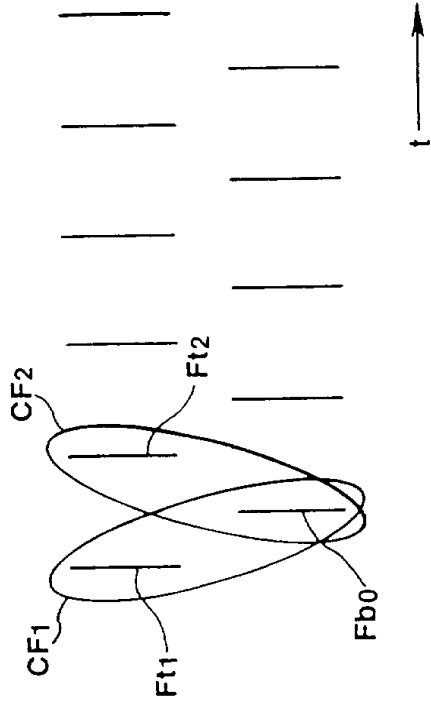


图 21A

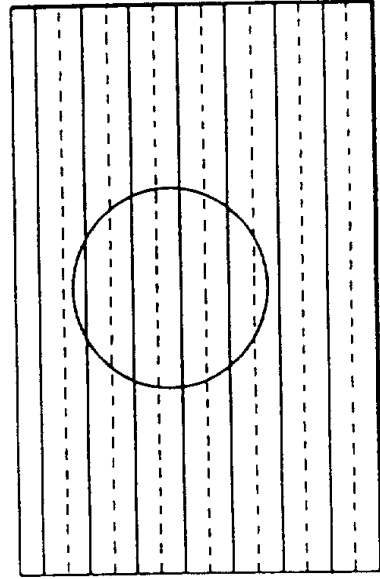


图 21B

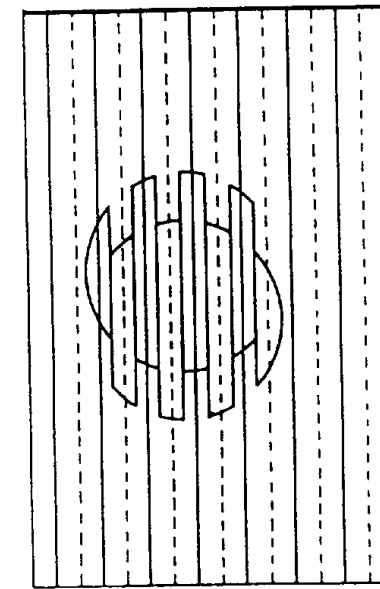


图 21C

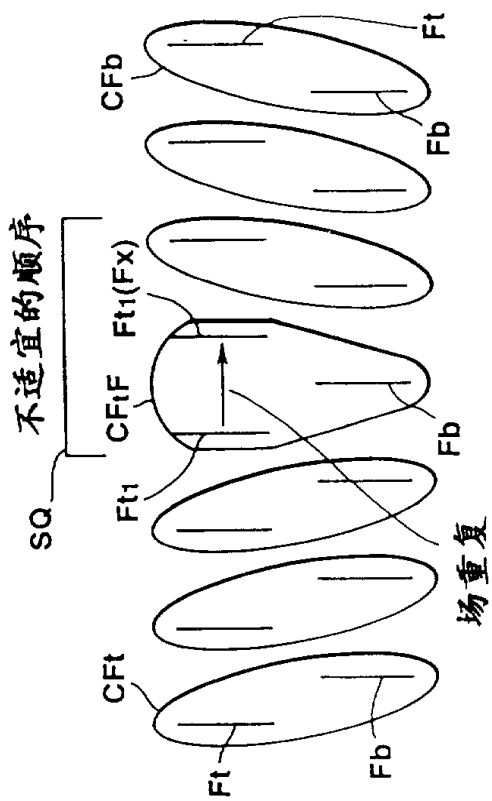


图 22C

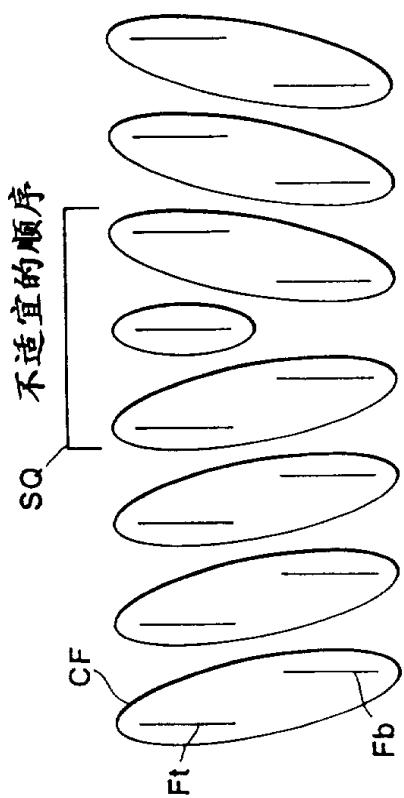


图 22A

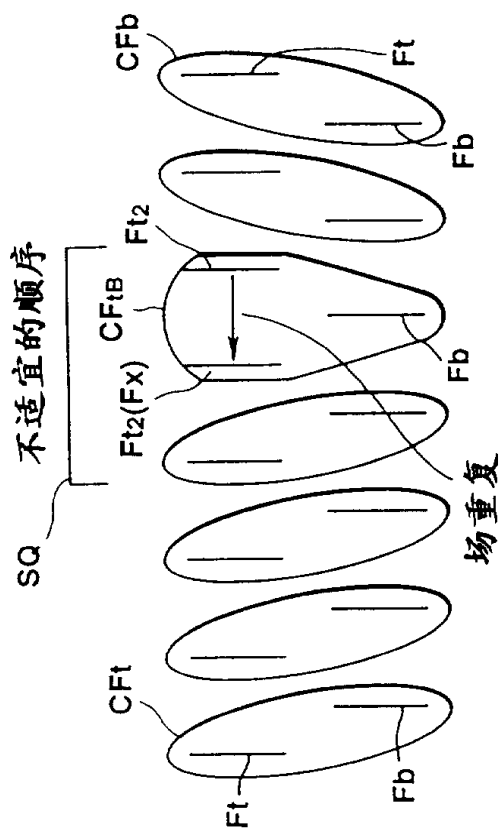


图 22D

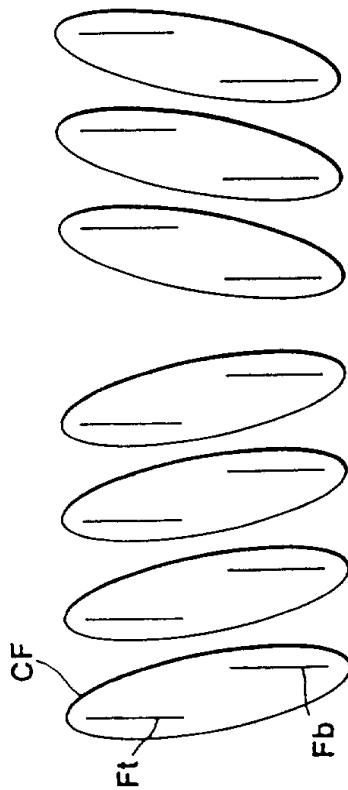


图 22B

top_field_first	1	1	1	1	1	0	0	0	0
repeat_first_field	0	0	0	1	1	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	↓	↓	↓	↓	↓	↓	↓	↓	↓

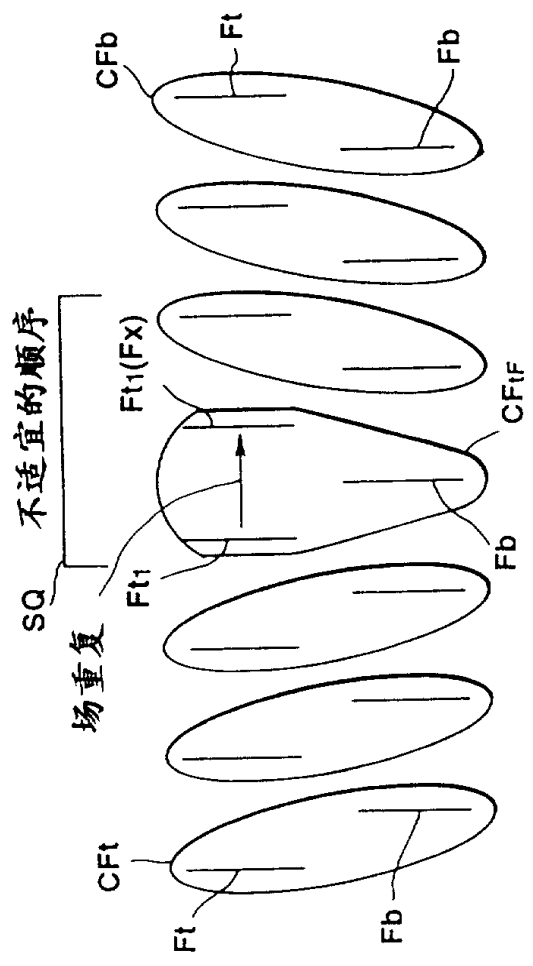


图 23

top_field_first	1	1	1	1	1	1	1	0	0
repeat_fiest_field	0	0	0	0	1	1	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	↓	↓	↓	↓	↓	↓	↓	↓	↓

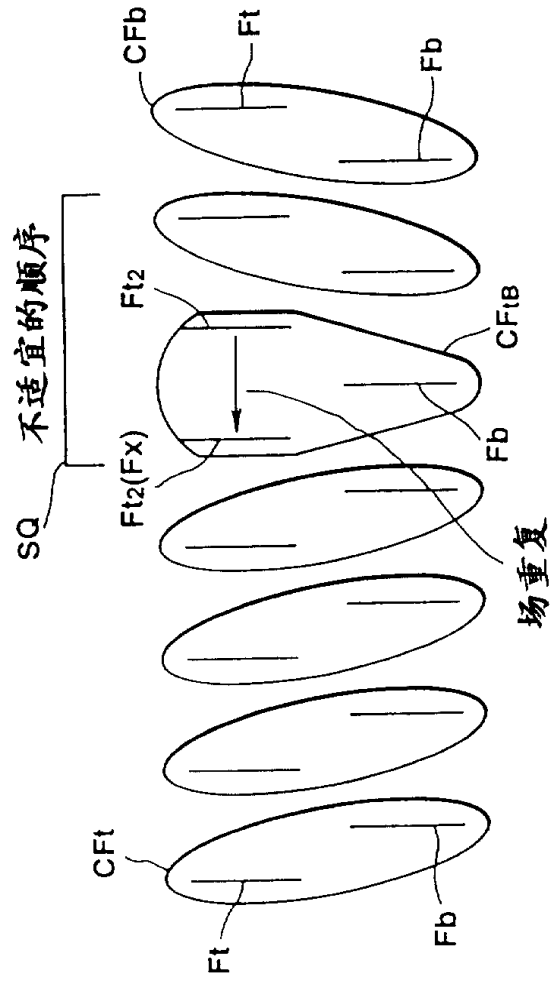


图 24

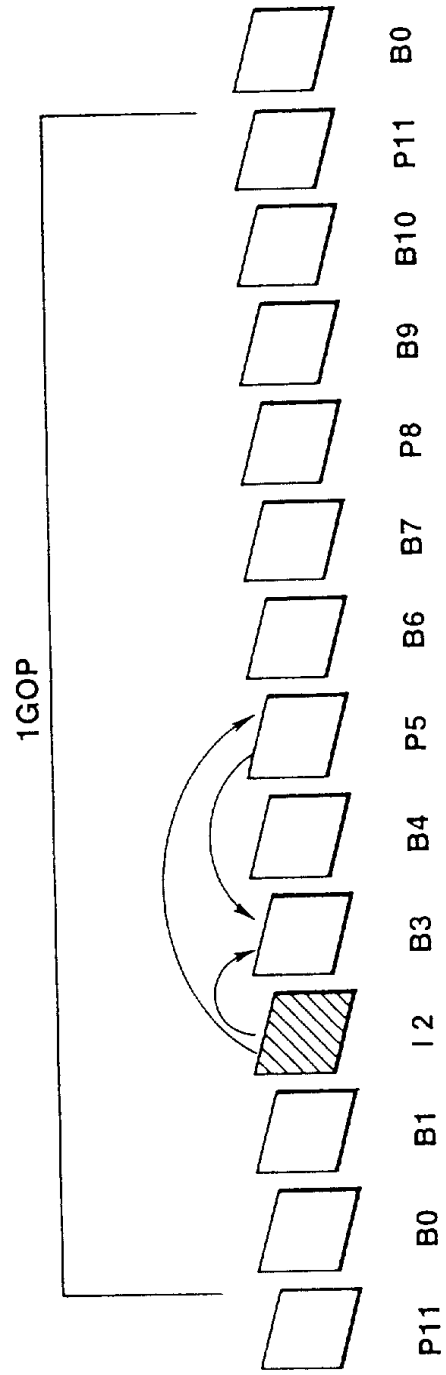


图 25

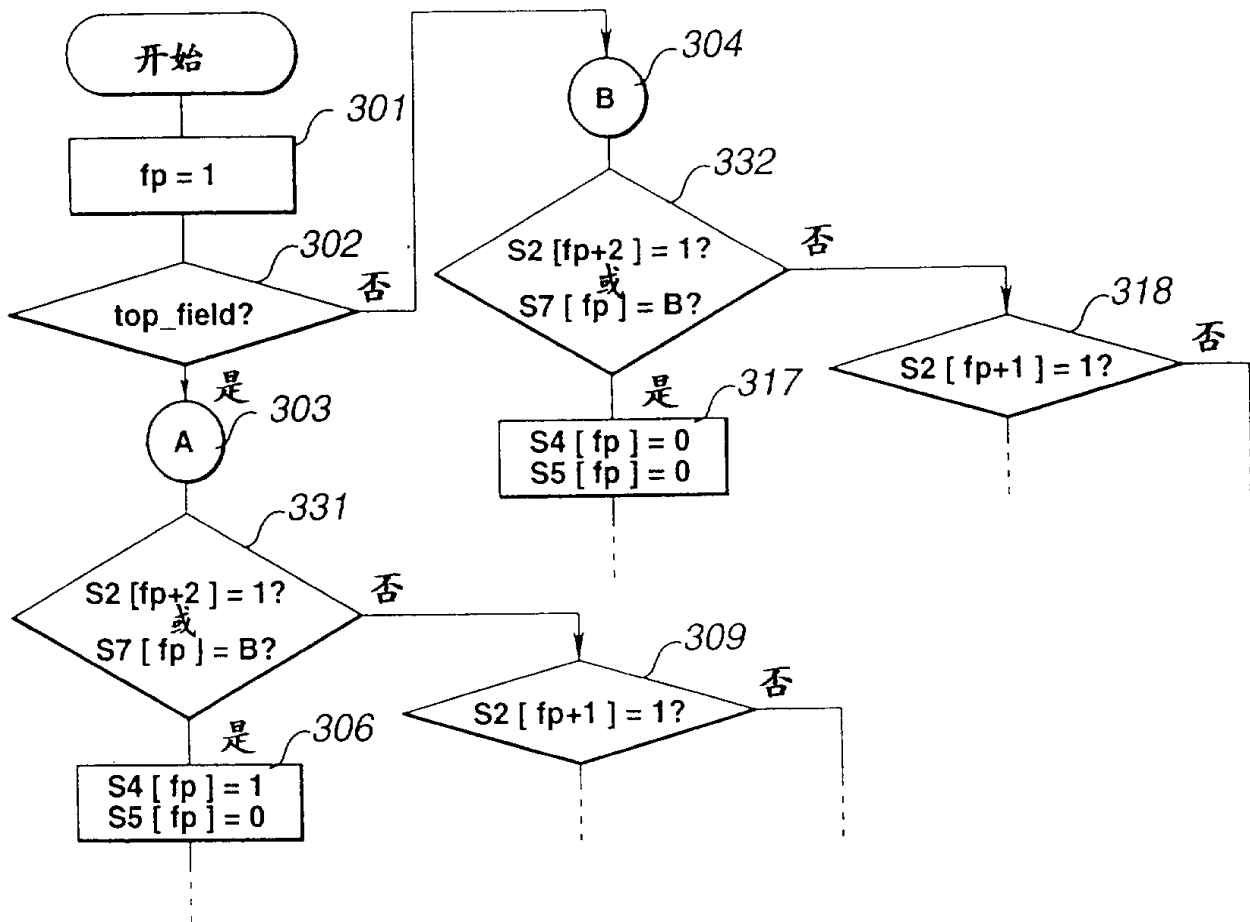


图 26

编辑点



S1	A	B	C	D	E	?	F	G	H	I
	a	b	c	d	e	f	g	h	i	
S2	1	0	1	0	1	0	1	0	1	1
	0	1	0	1	0	1	0	1	0	1
S3	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
S4	1	1	1	1	1	1	0	0	0	
	0	0	0	0	0	1	0	0	0	
S5	0	0	0	0	0	1	0	0	0	
	0	0	0	0	0	1	0	0	0	
S6	A	B	C	D	E		F	G	H	I
	a	b	c	d	e	f	g	h	i	
S7	I	B	B	P	B	B	P	B	B	
	I	B	B	P	B	B	P	B	B	

图 27A

编辑点



S1	A	B	C	D	E	?	F	G	H	I
	a	b	c	d	e	f	g	h	i	
S2	1	0	1	0	1	0	1	0	1	1
	0	1	0	1	0	1	0	1	0	1
S3	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
S4	1	1	1	1	1	1	1	0	0	
	0	0	0	0	0	1	1	0	0	
S5	0	0	0	0	0	1	1	0	0	
	0	0	0	0	0	1	1	0	0	
S6	A	B	C	D	E	?		G	H	I
	a	b	c	d	e	f	g	h	i	
S7	I	B	B	P	B	B	P	B	B	
	I	B	B	P	B	B	P	B	B	

图 27B

编辑点



S1	a	A	b	B	c	C	?	D	d	E	e	F	f	G	g	H	h
S2	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0
S3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
S4	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
S5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S6	a	A	b	B	c	C	D	E	e	F	f	G	g	H	h		
S7	B	I	B	B	B	P	B	B	P								

图 28A

编辑点



S1	a	A	b	B	c	C	?	D	d	E	e	F	f	G	g	H	h
S2	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0
S3	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
S4	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
S6	a	A	b	B	c	C	D	E	e	F	f	G	g	H	h		
S7	B	I	B	B	B	p	B	B	P								

图 28B

编辑点



S1	A	B	C	D	E	?	F	G	H	I
	a	b	c	d	e	f	g	h	i	
S2	1	0	1	0	1	0	1	0	1	1
	0	1	0	1	0	1	0	1	0	1
S3	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	0	0	0
S4	1	1	1	1	1	1	0	0	0	
S5	0	0	0	0	0	1	0	0	0	
S6	A	B	C	D	E	F	G	H	I	
	a	b	c	d	e	f	g	h	i	
	I	B	B	P	B	B	P	B	B	
S7	I	B	B	P	B	P	P	B	B	

图 29

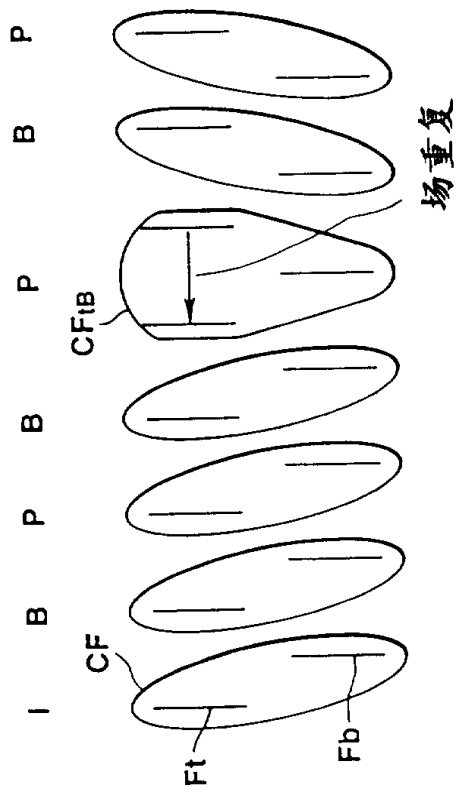


图 30B

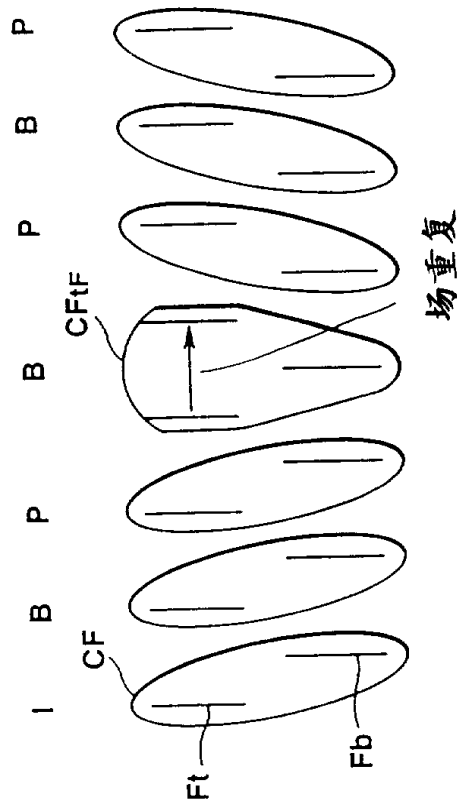


图 30C

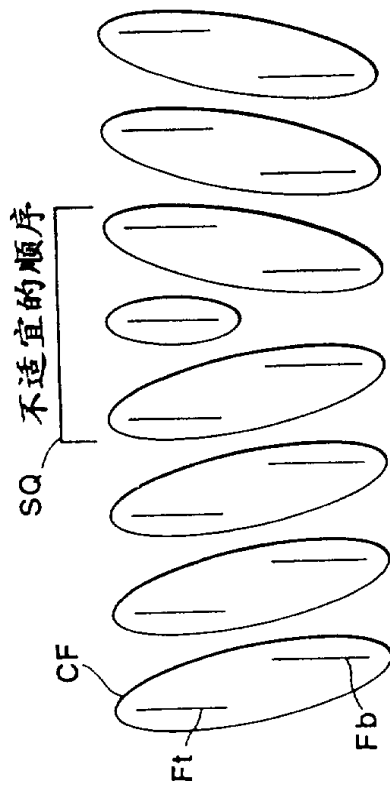


图 30A

编辑点



编辑点



S1	A	B	C	?	D	E	F	G	H	I	J	K
	a	b	c	d	e	f	g	?	h	i	j	k
S22	1	0	1	0	1	0	1	0	1	1	0	1
S27	I	B	B	P	B	B	P	B	B	P	B	B
S24(tff)	1	1	1	0	0	0	0	0	0	0	0	1
S25(tff)	0	0	0	1	0	0	0	0	0	0	1	0
模式	1	1	1	2	4	4	4	4	4	4	5	1
S103	A	B	C	?	D	E	F	G	H	I	J	K
	a	b	c	d	e	f	g	?	h	i	j	k
S104	A	B	C	?	D	E	F	G	H	I	J	K
	a	b	c	d	e	f	g	?	h	i	j	k
S105	A	B	C	?	D	E	F	G	H	I	J	K
	a	b	c	d	e	f	g	?	h	i	j	k
S26(下拉顺序)	A	B	C	D	-	E	F	G	H	I	J	K
	a	b	c	d	-	e	f	g	?	h	i	k
帧延迟	4	4	4	4	3	4	-	4	5	4	4	4

图 31

编辑点
↓

编辑点
↓

S1	A	B	C	?	D	E	F	G	H	I	J	K				
	a	b	c	d	e	f	g	?	h	i	j	k				
S22	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
S27	I	B	P	B	B	P	B	B	P	B	P	B				
S24(tff)	1	1	1	1	0	0	0	1	1	1	1	1				
S25(rff)	0	0	1	1	0	0	1	0	0	0	0	0				
模式	1	1	2	4	4	6	1	1	1	1	1					
S103	A	B	C	?	D	E	F	G	H	I	J	K				
	a	b	c	d	e	f	g	?	h	i	j	k				
S104	A	B	C	?	D	E	F	G	H	I	J	K				
	a	b	c	d	e	f	g	?	h	i	j	k				
S105	A	B	C	?	D	E	F	G	H	I	J	K				
	a	b	c	d	e	f	g	?	h	i	j	k				
S26	A	B	C	D	—	E	F	G	H	I	J	K				
	a	b	c	d	—	e	f	g	h	i	j	k				
帧延迟	4	4	4	4	3	4	—	4	5	4	5	4	4	4	4	4

图 32

```

1  function ( ) {
2      n = 0;
3      isTop_field = 1;
4
5      while (!EOF) {
6
7          if (picture_coding_type != B_PICTURE &&
8              S22[n+3] == 1 &&
9              S22[n+4] == 0 &&
10             S22[n+5] == 1 &&
11             S22[n+6] == 0 {
12
13             if (isTop_field == 1) {
14                 if (S22[n+1] == 1) {
15                     top_field_first[n] = 1, repeat_first_field[n] = 1, pattern[n] = 2;
16                 } else
17                     top_field_first[n] = 1, repeat_first_field[n] = 1, pattern[n] = 3;
18
19                 isTop_field = 0;
20             }
21             else {
22                 if (S22[n+1] == 1)
23                     top_field_first[n] = 0, repeat_first_field[n] = 1, pattern[n] = 5;
24                 else
25                     top_field_first[n] = 0, repeat_first_field[n] = 1, pattern[n] = 6;
26
27                 isTop_field = 1;
28             }
29
30             n = n+3;
31
32         }
33     else {
34
35         if (isTop_field == 1)
36             top_field_first[n] = 1, repeat_first_field[n] = 0, pattern[n] = 1;
37         else
38             top_field_first[n] = 0, repeat_first_field[n] = 0, pattern[n] = 4;
39
40         n = n+2;
41
42     }
43 }
44
45 }

```

图 33

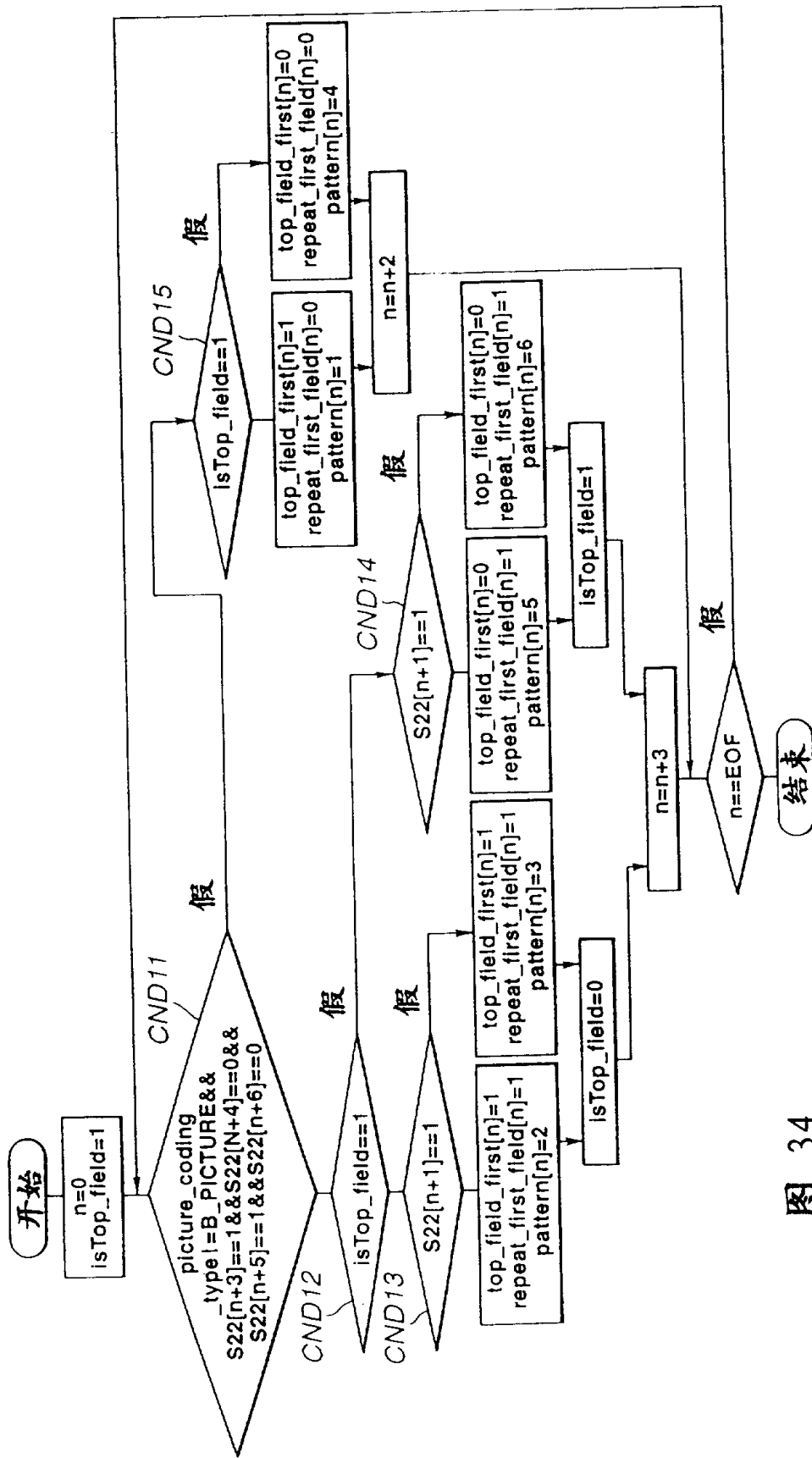


图 34

S207	A	B	C	D	E	F	
	a	b	c	d	e	f	
S10	1	1	1	1	0	0	
S11	0	0	0	1	0	0	
S208	A	B	C	D	D	E	F
	a	b	c	d	e	f	

图 35

S207	A	B	C	D	E	F	
	a	b	c	d	e	f	
S10	1	1	1	0	0	0	
S11	0	0	1	0	0	0	
S208	A	B	C	C	D	E	F
	a	b	c	d	e	f	

图 36

S207	e	E	f	F	g	G	h	H	i	J	j	K	k
S10	0	0	0	0	0	0	0	0	1	1	1	1	1
S11	0	0	0	0	1	0	0	0	0	0	0	0	0
S208	e	E	f	F	g	G	h	H	i	J	j	K	k

图 37

S207	e	E	f	F	g	G	h	H	i	J	j	K	k
S10	0	0	0	0	0	0	0	0	1	1	1	1	1
S11	0	0	0	1	0	0	0	0	0	0	0	0	0
S208	e	E	f	F	g	G	h	H	i	J	j	K	k

图 38

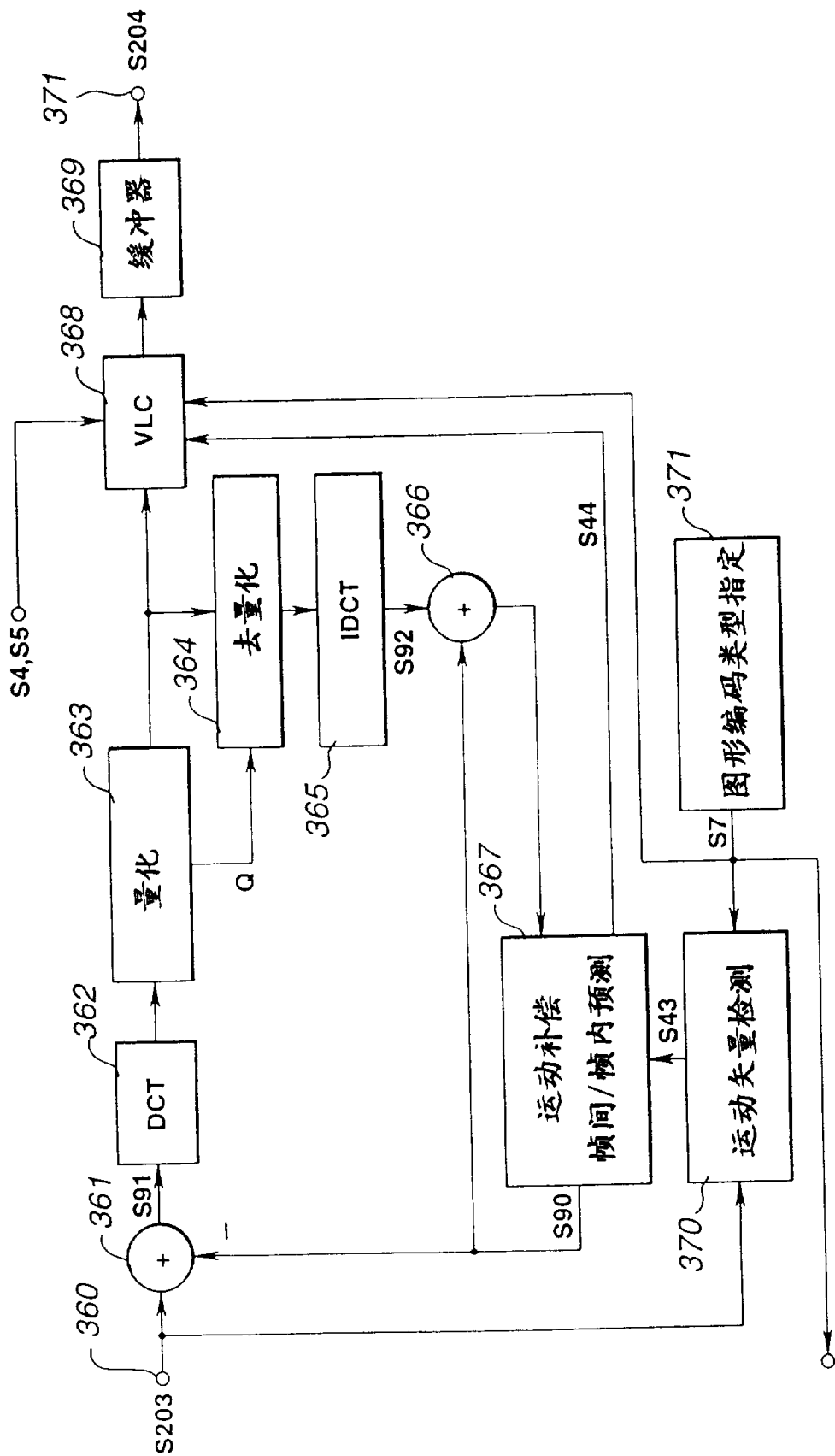


图 39

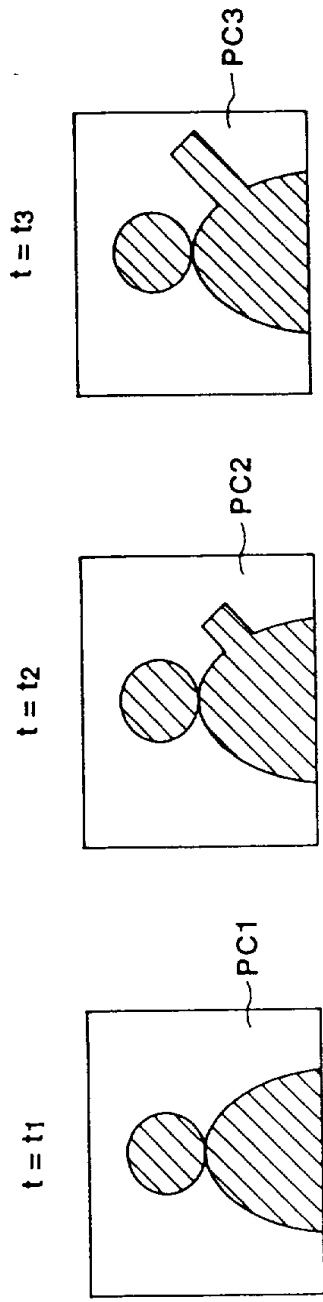


图 40A

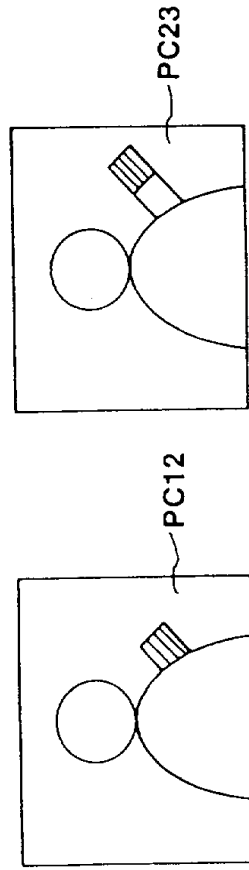


图 40B

图形组

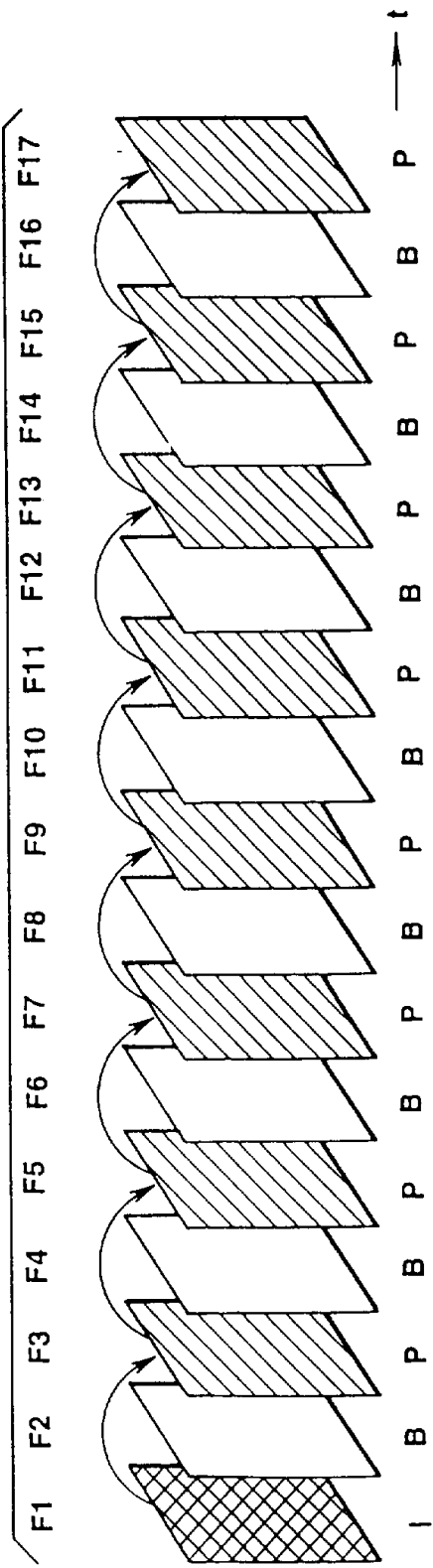


图 41A

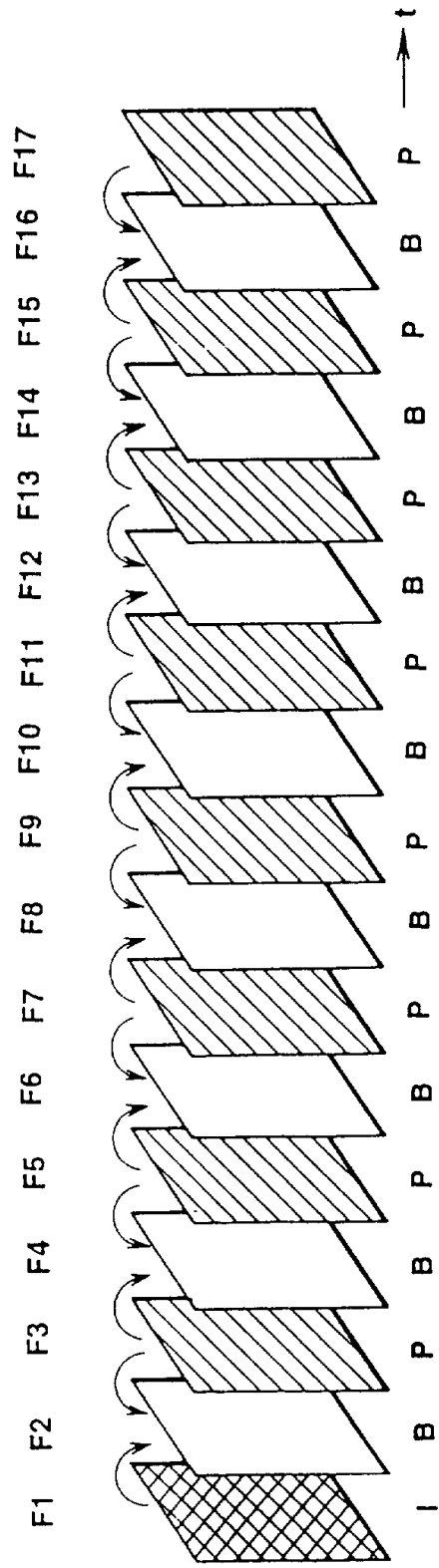


图 41B

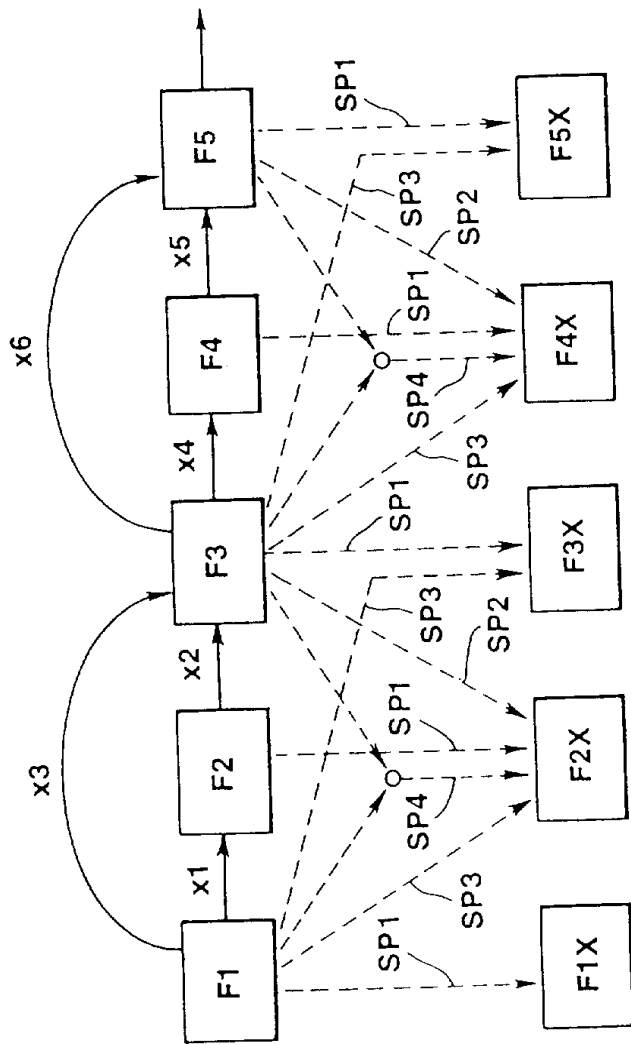


图 42A

图 42B

I B P B P B P

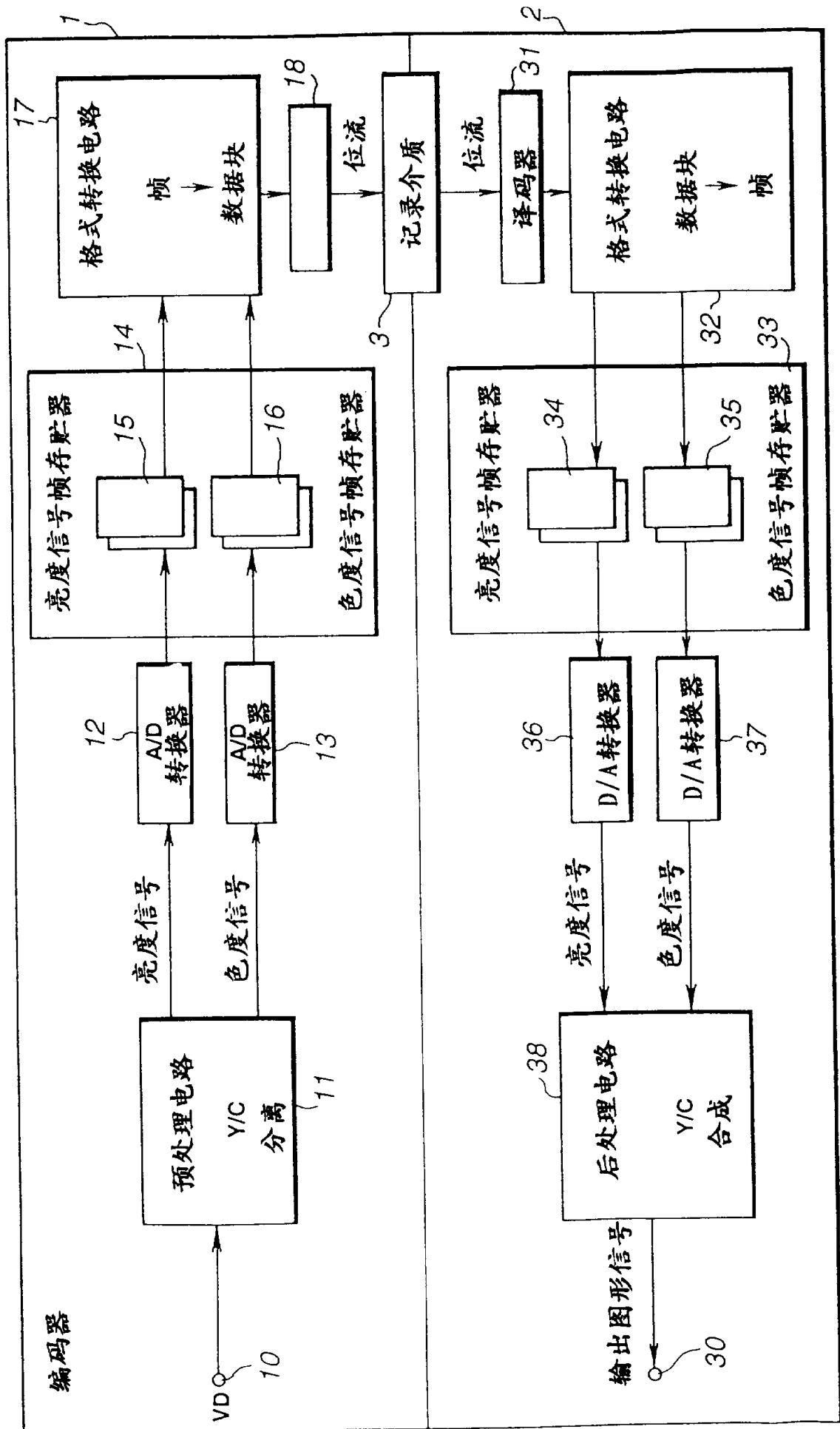


图 43

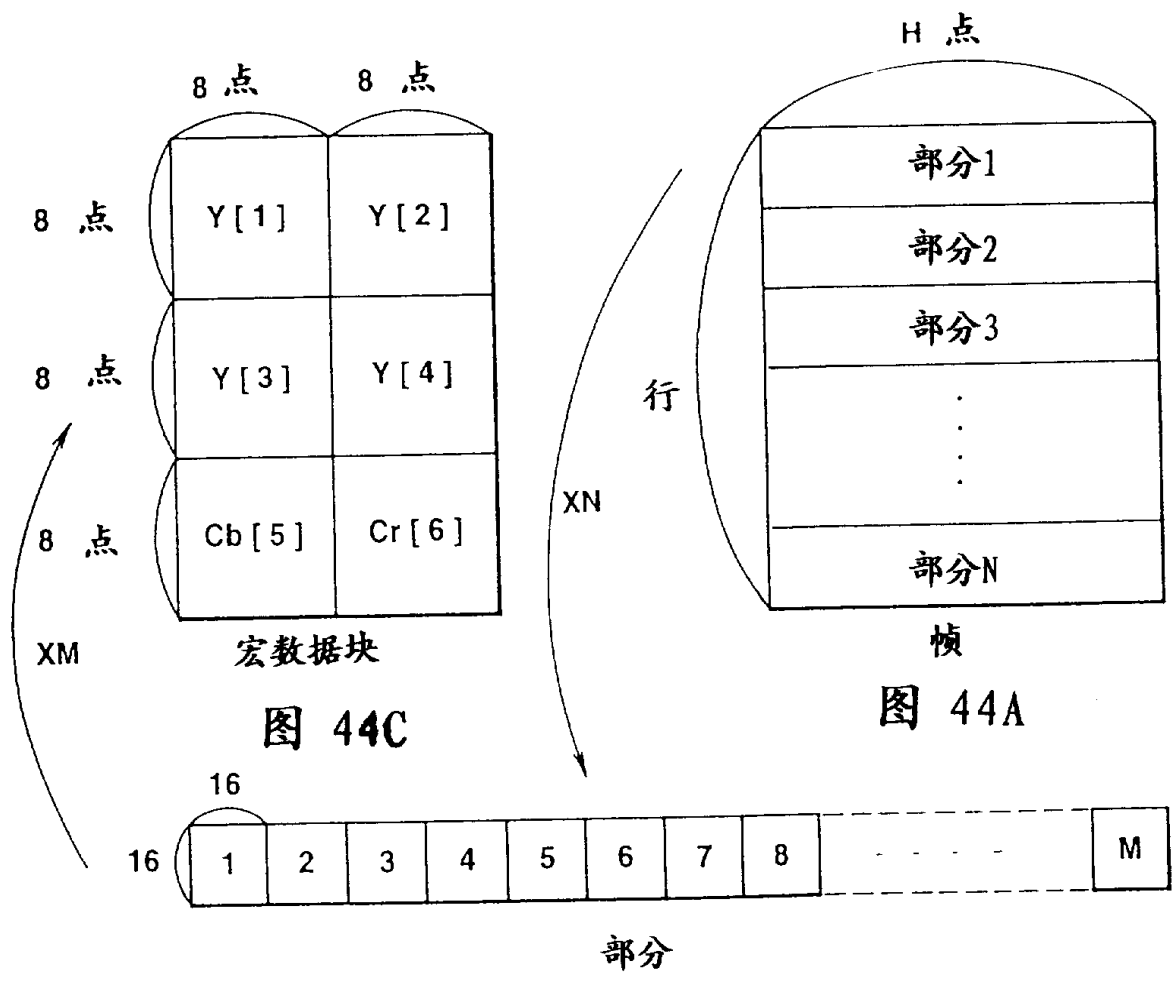


图 44B

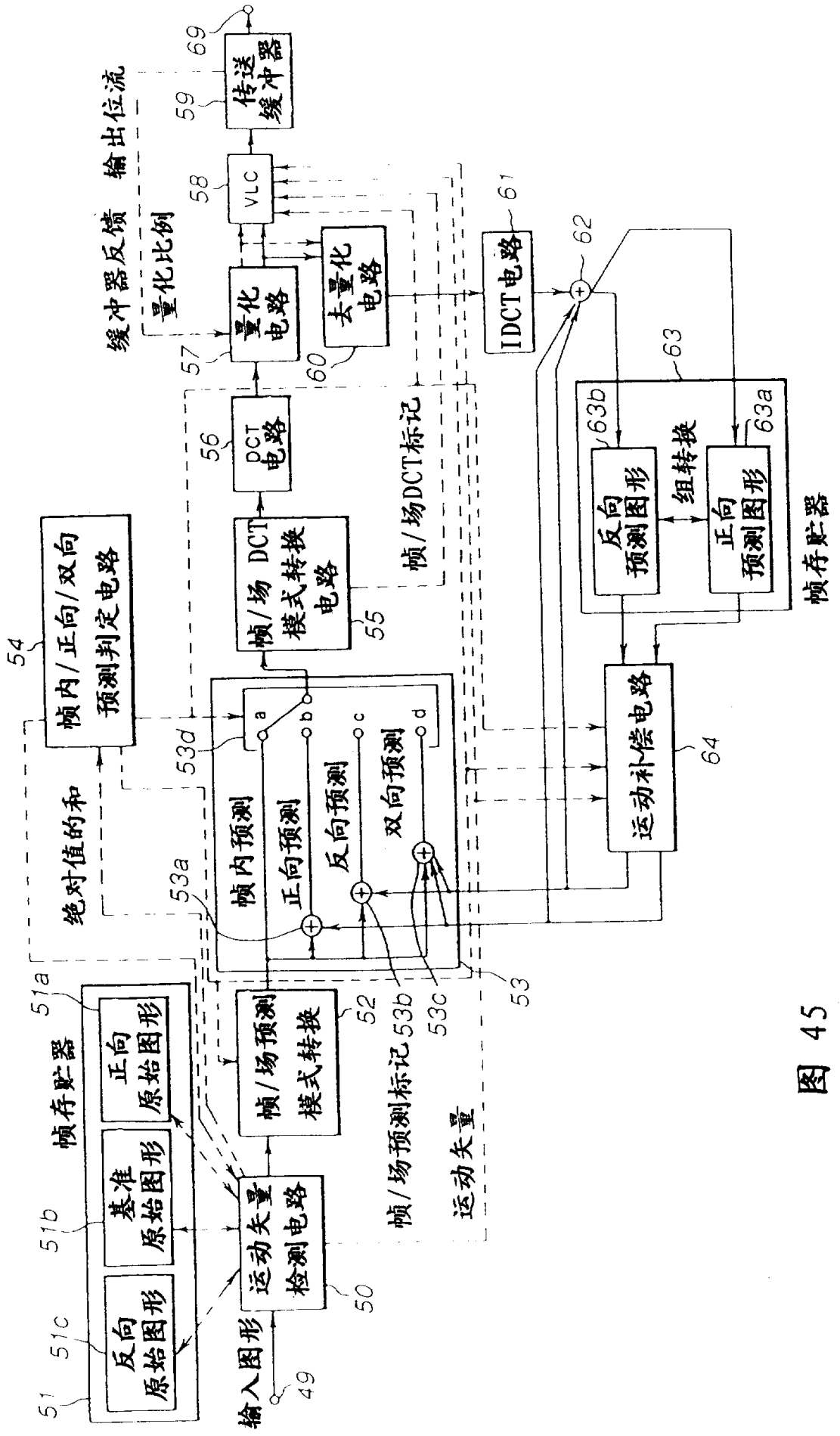
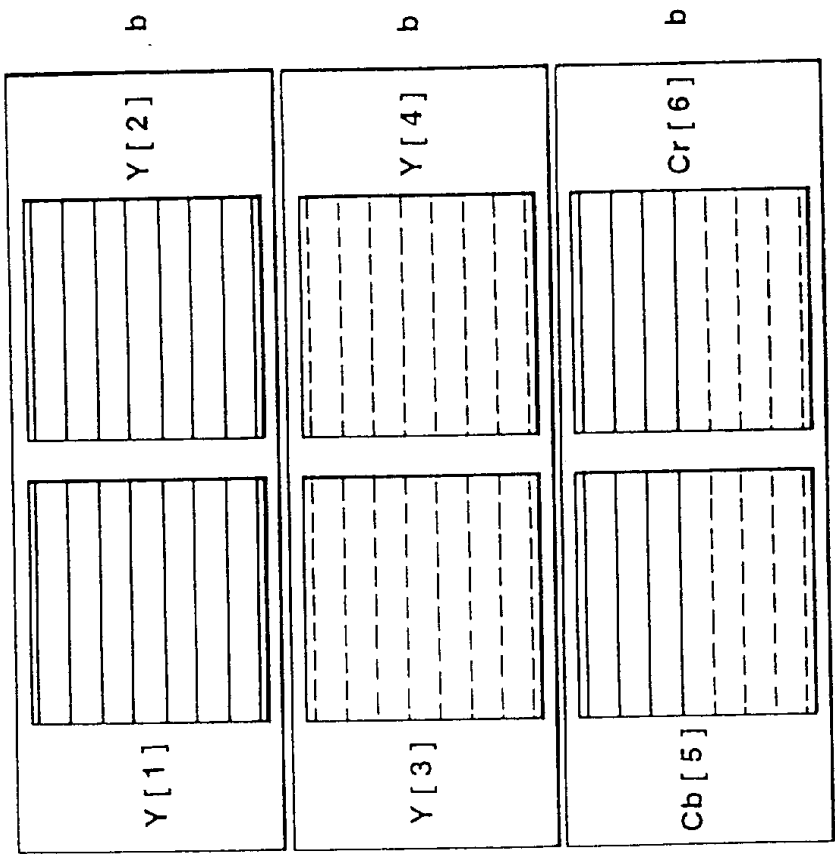
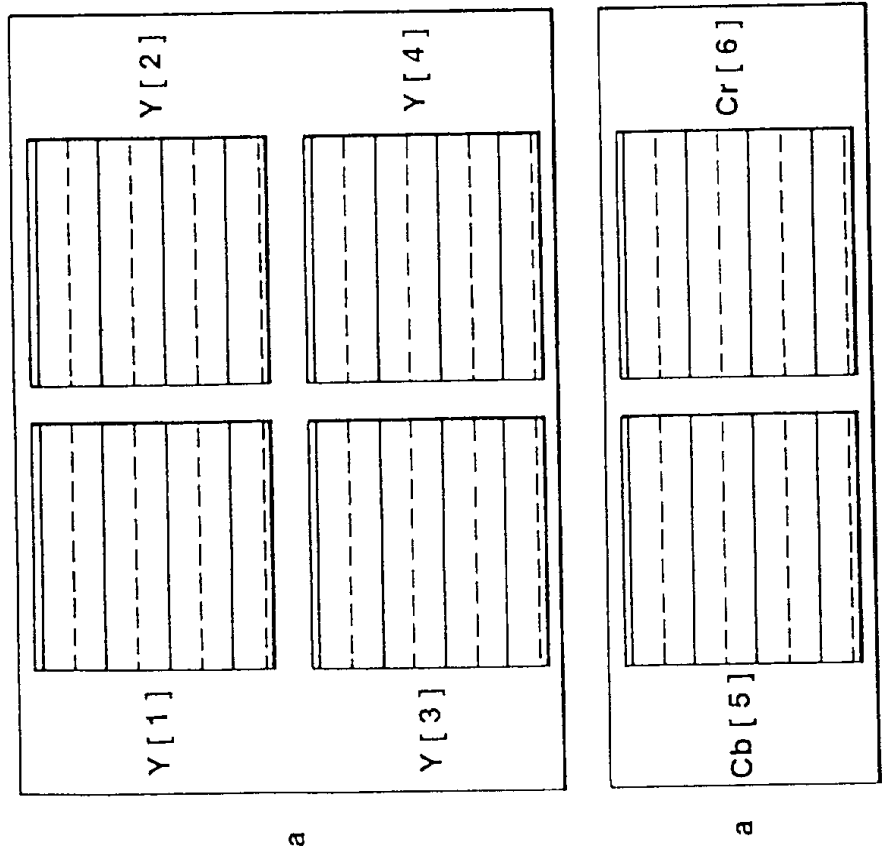


图 45



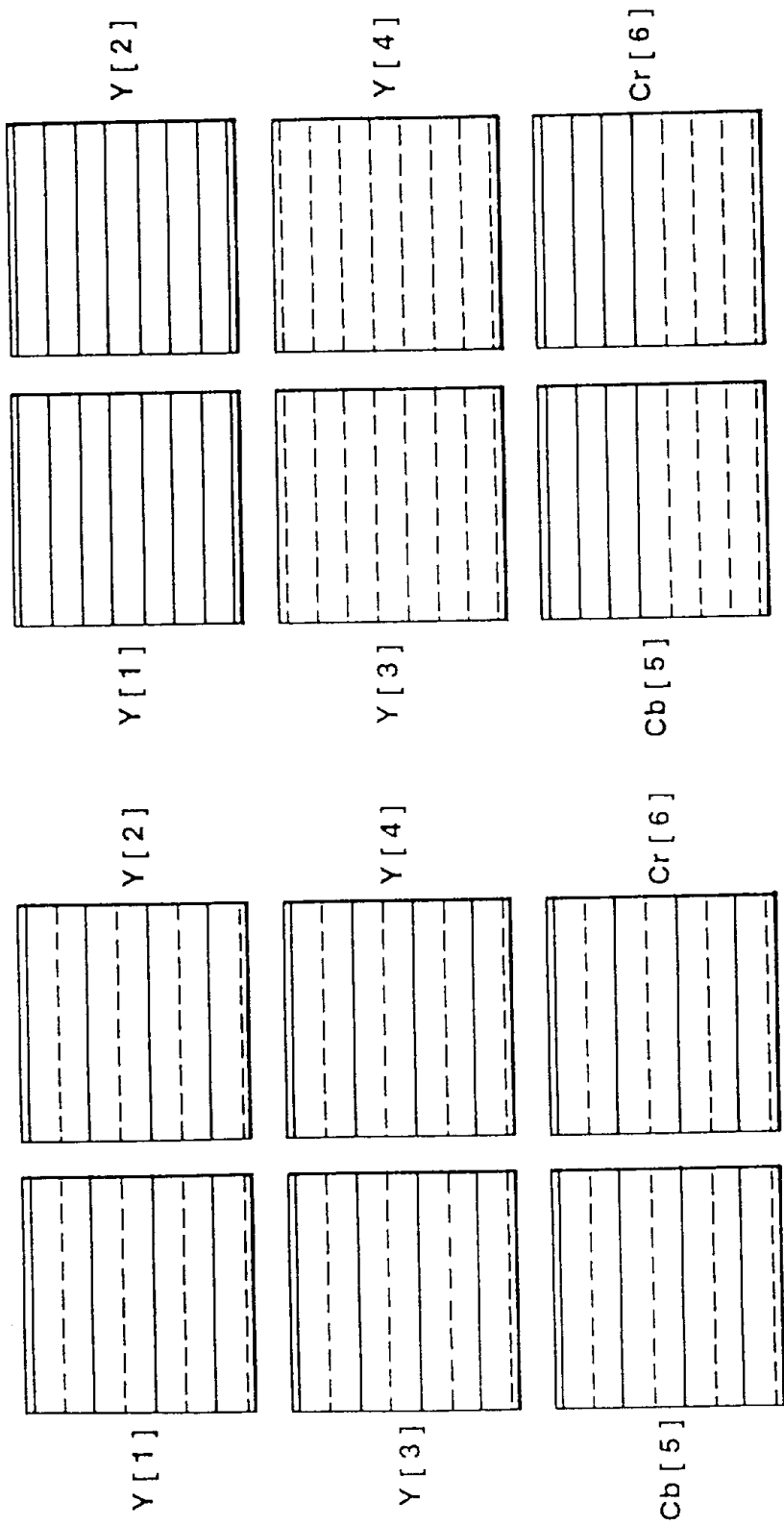
场预测模式

图 46B



帧预测模式

图 46A



场DCT模式

帧DCT模式

图 47B

图 47A

90 译码电路

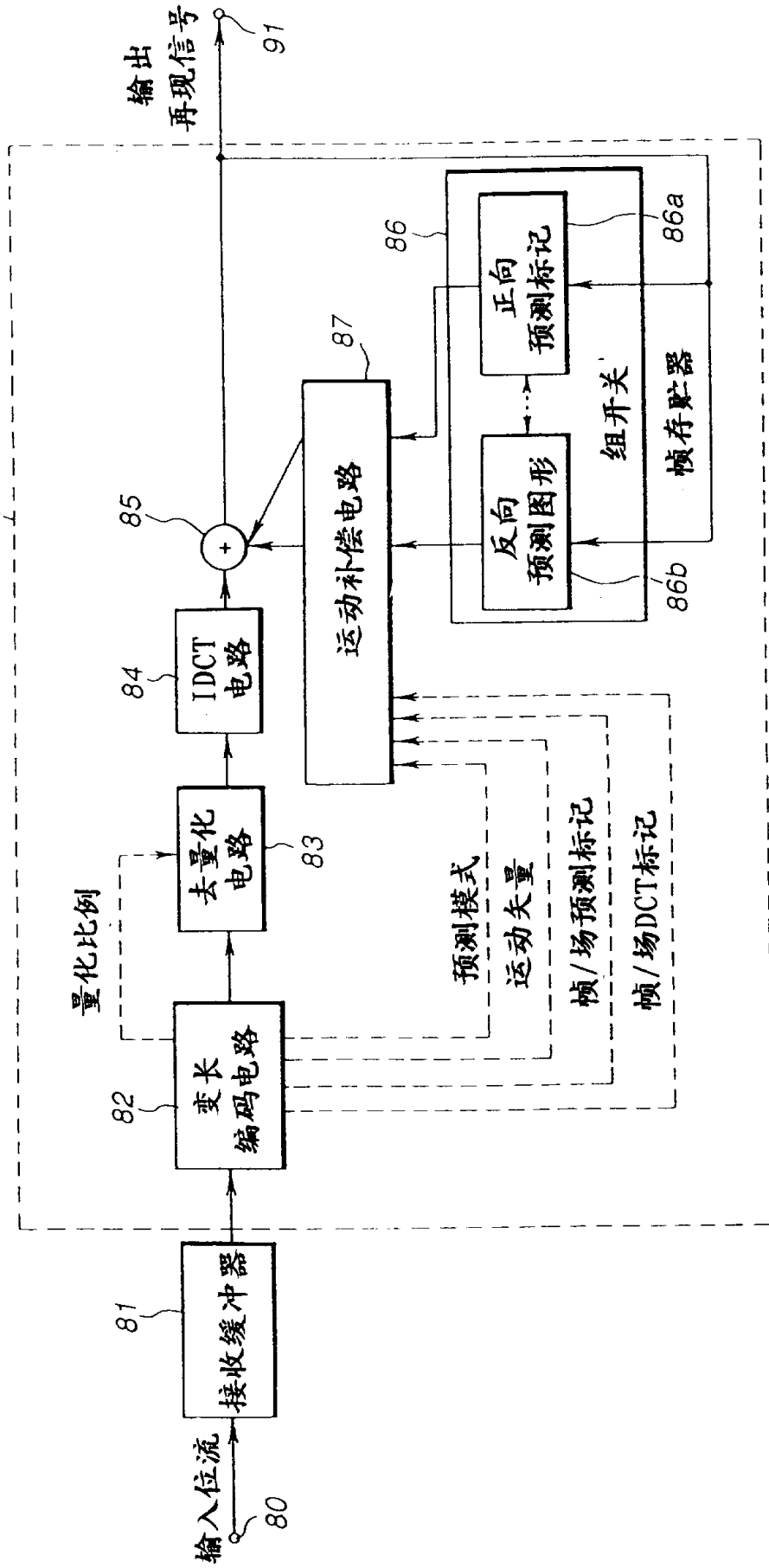


图 48