



(12) 发明专利

(10) 授权公告号 CN 101202095 B

(45) 授权公告日 2012. 05. 30

(21) 申请号 200710305122. 1

CN 1647184 A, 2005. 07. 27,

(22) 申请日 2007. 06. 06

US 2001028612 A1, 2001. 10. 11,

(30) 优先权数据

审查员 桂煦

2006-157762 2006. 06. 06 JP

(73) 专利权人 索尼株式会社

地址 日本东京

(72) 发明人 内村幸一

(74) 专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 李镇江

(51) Int. Cl.

H04N 9/804 (2006. 01)

G11B 27/10 (2006. 01)

G11B 27/34 (2006. 01)

G11B 27/32 (2006. 01)

G11B 19/02 (2006. 01)

G11B 27/034 (2006. 01)

(56) 对比文件

CN 1335617 A, 2002. 02. 13,

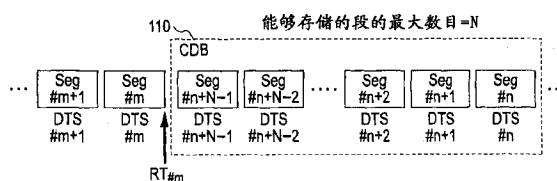
权利要求书 4 页 说明书 36 页 附图 35 页

(54) 发明名称

创作、记录、和验证设备、方法和程序

(57) 摘要

本发明提供了一种创作、记录、和验证设备、方法和程序。一种数据结构,包括内容数据和基于多个第一单元形成的伴随数据,该伴随数据用于显示伴随内容数据的信息。在定义时间信息的第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。



1. 一种用于产生包括内容数据和基于多个第一单元形成的伴随数据的数据结构的创作设备,该伴随数据用于显示伴随该内容数据的信息,该创作设备包括:

产生内容数据和伴随数据的数据产生器;

对数据产生器所产生的内容数据和伴随数据进行编码的编码器;以及

对编码器编码后的内容数据和伴随数据进行多路复用的多路复用器,

其中,编码器所执行的编码和多路复用器所执行的多路复用使得,在所述数据结构中,在定义时间信息的多个第一单元中,目标第一单元被提供给缓冲器的时间被设置得晚于解码时间,其中该缓冲器是为用于解码第一单元的解码器而提供的,所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义,其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限。

2. 如权利要求 1 所述的创作设备,其中,如果为第一单元定义解码时间,则根据解码时间排列第一单元。

3. 如权利要求 1 所述的创作设备,其中,在第一单元中,定义关于伴随数据的结构的结构信息、关于伴随数据的属性的属性信息、关于伴随数据的图像信息、或关于伴随数据的分割点的分割点信息,

从包括一项结构信息和一项分割点信息的多个第一单元形成第二单元,

从包括一项或多项属性信息的一个或多个所述第二单元形成第三单元,

在第三单元中,存储在第一单元中定义的结构信息和属性信息,并且当第三单元中包括在第一单元中定义的图像信息时,在第三单元中存储该图像信息,并且

伴随数据是一组第三单元。

4. 如权利要求 1 所述的创作设备,其中在包括第一单元的包的包头部中存储关于解码时间的信息。

5. 如权利要求 1 所述的创作设备,其中伴随数据包括将通过覆盖在由内容数据所表示的主图像上而显示的图形数据,在图形数据中定义输出时间。

6. 如权利要求 1 所述的创作设备,其中伴随数据包括按钮图像,该按钮图像实现动画显示,并用于显示指示用户执行操作的操作屏幕所使用的按钮,而且该伴随数据还包括关于按钮图像的显示控制信息。

7. 如权利要求 1 所述的创作设备,其中为存储结构信息或图像信息的第一单元定义关于解码时间的信息。

8. 如权利要求 7 所述的创作设备,其中还为存储属性信息的第一单元定义关于解码时间的信息。

9. 一种用于产生包括内容数据和基于多个第一单元形成的伴随数据的数据结构的创作方法,该伴随数据用于显示伴随该内容数据的信息,该创作方法包括步骤:

产生内容数据和伴随数据;

对所产生的内容数据和伴随数据进行编码;以及

对编码后的内容数据和伴随数据进行多路复用,

其中,产生所述数据结构,使得在定义时间信息的多个第一单元中,目标第一单元被提供给缓冲器的时间被设置得晚于解码时间,其中该缓冲器是为用于解码第一单元的解码器而提供的,所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定

义,其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限。

10. 一种记录设备,用于在记录介质上记录包括内容数据和基于多个第一单元形成的伴随数据的数据结构,该伴随数据用于显示伴随该内容数据的信息,该记录设备包括:

产生内容数据和伴随数据的数据产生器;

对数据产生器所产生的内容数据和伴随数据进行编码的编码器;

对编码器编码后的内容数据和伴随数据进行多路复用的多路复用器;

转换器,用于将多路复用器多路复用后的内容数据和伴随数据转换为适合在记录介质上记录的格式;以及

记录器,用于将转换器转换后的内容数据和伴随数据记录在记录介质上,

其中,编码器所执行的编码和多路复用器所执行的多路复用使得,在所述数据结构中,在定义时间信息的多个第一单元中,目标第一单元被提供给缓冲器的时间被设置得晚于解码时间,其中该缓冲器是为用于解码第一单元的解码器而提供的,所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义,其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限。

11. 如权利要求 10 所述的记录设备,其中,如果为第一单元定义解码时间,则根据解码时间排列第一单元。

12. 如权利要求 10 所述的记录设备,其中,在第一单元中,定义关于伴随数据的结构的结构信息、关于伴随数据的属性的属性信息、关于伴随数据的图像信息、或关于伴随数据的分割点的分割点信息,

从包括一项结构信息和一项分割点信息的多个第一单元形成第二单元,

从包括一项或多项属性信息的一个或多个所述第二单元形成第三单元,

在第三单元中,存储在第一单元中定义的结构信息和属性信息,并且当第三单元中包括在第一单元中定义的图像信息时,在第三单元中存储该图像信息,并且

伴随数据是一组第三单元。

13. 如权利要求 10 所述的记录设备,其中在包括第一单元的包的包头部中存储关于解码时间的信息。

14. 如权利要求 10 所述的记录设备,其中伴随数据包括将通过覆盖在由内容数据所表示的主图像上而显示的图形数据,在图形数据中定义输出时间。

15. 如权利要求 10 所述的记录设备,其中伴随数据包括按钮图像,该按钮图像实现动画显示,并用于显示指示用户执行操作的操作屏幕所使用的按钮,而且伴随数据还包括关于按钮图像的显示控制信息。

16. 如权利要求 10 所述的记录设备,其中为存储结构信息或图像信息的第一单元定义关于解码时间的信息。

17. 如权利要求 16 所述的记录设备,其中还为存储属性信息的第一单元定义关于解码时间的信息。

18. 一种记录方法,用于在记录介质上记录包括内容数据和基于多个第一单元形成的伴随数据的数据结构,该伴随数据用于显示伴随该内容数据的信息,该记录方法包括步骤:

产生内容数据和伴随数据；

对所产生的内容数据和伴随数据进行编码；

对编码后的内容数据和伴随数据进行多路复用；

将多路复用后的内容数据和伴随数据转换为适合在记录介质上记录的格式；以及

将转换后的内容数据和伴随数据记录在记录介质上，

其中在记录介质上记录这样的数据结构：在定义时间信息的多个第一单元中，目标第一单元被提供给缓冲器的时间被设置得晚于解码时间，其中该缓冲器是为用于解码第一单元的解码器而提供的，所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义，其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限。

19. 一种验证设备，用于执行对数据结构的验证，该数据结构包括内容数据和基于多个第一单元形成的伴随数据，该伴随数据用于显示伴随该内容数据的信息，

其中，在该数据结构中，在定义时间信息的多个第一单元中，目标第一单元被提供给缓冲器的时间被设置得晚于解码时间，其中该缓冲器是为用于解码第一单元的解码器而提供的，所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义，其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限，且输入具有该数据结构的数据，

该验证设备包括确定单元，所述确定单元用于针对所述数据确定在定义时间信息的多个第一单元中，目标第一单元被提供给缓冲器的时间是否被设置得晚于解码时间，其中该缓冲器是为用于解码第一单元的解码器而提供的，所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义，其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限。

20. 如权利要求 19 所述的验证设备，其中，如果为第一单元定义解码时间，则根据解码时间排列第一单元。

21. 如权利要求 19 所述的验证设备，其中，在第一单元中，定义关于伴随数据的结构的信息、关于伴随数据的属性的属性信息、关于伴随数据的图像信息、或关于伴随数据的分割点的分割点信息，

从包括一项结构信息和一项分割点信息的多个第一单元形成第二单元，

从包括一项或多项属性信息的一个或多个所述第二单元形成第三单元，

在第三单元中，存储在第一单元中定义的结构信息和属性信息，并且当第三单元中包括在第一单元中定义的图像信息时，在第三单元中存储该图像信息，并且

伴随数据是一组第三单元。

22. 如权利要求 19 所述的验证设备，其中确定单元对第一单元中的每一个执行确定。

23. 如权利要求 19 所述的验证设备，其中在包括第一单元的包的包头中存储关于解码时间的信息。

24. 如权利要求 19 所述的验证设备，其中伴随数据包括将通过覆盖在由内容数据所表示的主图像上而显示的图形数据，在图形数据中定义输出时间。

25. 如权利要求 19 所述的验证设备，其中伴随数据包括按钮图像，该按钮图像实现动画显示，并用于显示指示用户执行操作的操作屏幕所使用的按钮，而且伴随数据还包括关

于按钮图像的显示控制信息。

26. 如权利要求 19 所述的验证设备,其中为存储结构信息或图像信息的第一单元定义关于解码时间的信息。

27. 如权利要求 26 所述的验证设备,其中还为存储属性信息的第一单元定义关于解码时间的信息。

28. 一种验证方法,用于执行对数据结构的验证,该数据结构包括内容数据和基于多个第一单元形成的伴随数据,该伴随数据用于显示伴随该内容数据的信息,

其中,在该数据结构中,在定义时间信息的多个第一单元中,目标第一单元被提供给缓冲器的时间被设置得晚于解码时间,其中该缓冲器是为用于解码第一单元的解码器而提供的,所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义,其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限,且输入具有该数据结构的数据,

该验证方法包括针对所述数据确定在定义时间信息的多个第一单元中,目标第一单元被提供给缓冲器的时间是否被设置得晚于解码时间的步骤,其中该缓冲器是为用于解码第一单元的解码器而提供的,所述解码时间在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义,其中 N 表示所述为用于解码第一单元的解码器而提供的缓冲器中可存储的第一单元的数目的上限。

创作、记录、和验证设备、方法和程序

[0001] 对相关申请的交叉引用

[0002] 本发明包含与 2006 年 6 月 6 日向日本专利局提交的日本专利申请 JP2006-157762 相关的主题,在此引入其全部内容作为参考。

[0003] 技术领域

[0004] 本发明涉及一种数据结构,该数据结构允许在回放设备中回放对每一预定单元指定解码时间的数据,而不会干扰解码器缓冲器中所进行的处理,并且本发明还涉及一种记录该数据结构的记录介质。本发明还涉及一种用于产生该数据结构的创作设备、方法和程序,一种用于将该数据结构记录在记录介质上的记录设备、方法和程序,一种用于验证所产生数据是否满足特定条件的验证设备、方法和程序,以及一种用于制造记录介质的制造设备和方法。

[0005] 背景技术

[0006] 作为可以从记录 / 回放设备上拆卸的盘式记录介质的标准,提出了蓝光光盘的标准。根据蓝光光盘的标准,使用具有 0.1mm 的覆盖层的 12-cm 光盘作为记录介质,且使用具有 405nm 波长的蓝紫光激光器和 0.85 的数值孔径的物镜,作为光学系统,以便实现最大记录容量为 27 吉字节 (GB)。这将使得能记录多于 2 小时的日本标准广播卫星 (BS) 数字高清晰度 (HD) 节目,而不会降低图像质量成为可能。

[0007] 作为提供要记录到上述光盘上的音频 / 视频 (AV) 信号的源,采用两类的源:一种类型是用于提供通过模拟电视广播传送的模拟信号的源,另一种类型是用于提供数字电视广播传送的数字信号的源。根据蓝光光盘标准,已经定义了用于通过两种类型的广播系统记录 AV 信号的标准。

[0008] 作为当前蓝光光盘的衍生标准,开发出了记录电影或音乐的只读记录介质。作为用于记录电影或音乐的盘式记录介质,数字通用光盘 (DVD) 已经被广泛应用。然而,基于蓝光光盘标准的只读光盘比已知的 DVD 更优秀,因为它们能够通过利用蓝光光盘独特的特性,诸如大容量和高传输速度,在保持高图像质量的同时,记录多于 2 小时的 HD 视频。

[0009] 下文中,基于蓝光光盘的只读记录介质的标准,将被称为蓝光光盘只读存储格式 (BD-ROM) 标准。

[0010] 对于记录在只读记录介质上的内容来说,不可或缺的因素之一是字幕。在 BD-ROM 标准中,在与显示活动图像的面板不同的面板上显示字幕,且通过将字幕面板和活动图像面板相组合,通过彼此的重叠,显示字幕和活动图像。例如,将字幕面板覆盖在活动图像面板之上,且在字幕面板中,除字幕之外的部分被设置为透明。然后,形成一幅图像数据,其中在活动图像上显示字幕。

[0011] 为了在光盘上记录内容,诸如电影,且提供作为封装记录介质销售的光盘,用于控制伴随内容的不同程序的执行的用户界面与内容一起记录在光盘上。一个典型的用户界面是屏幕上显示的菜单。作为屏幕上所显示菜单的一个实例,作为按钮图像,提供用于选择功能的按钮,且通过预定输入单元,用户能够选择并设置多个按钮之一,然后,执行分配给选定按钮的功能。通常,通过使用与播放器兼容的远程控制器所提供的横向键,用户选择显示

在屏幕上的按钮,并按压设定键。然后,执行与该按钮相关的功能。

[0012] 日本来审专利申请公开号 2004-304767 公开了下列技术。在该公开文献所公开的技术中,基于记录/回放标准,即,蓝光光盘标准(蓝光光盘可改写格式第 1.0 版),分别提供显示活动图像的活动图像面板和显示字幕的字幕面板。然后,基于视频数据的活动图像和基于字幕图像的字幕一起显示在一个屏幕上。此外,还公开了利用按钮图形在屏幕上显示菜单的技术。

[0013] 字幕图像数据或菜单屏幕显示包括关于数据的结构的信息,关于数据的属性的信息,实际数据,以及表示数据的结束的信息,它们根据创作者的意愿而确定的分割点进行划分。在上述数据项的每一个中,指定解码时间。也可以指定输出时间。也就是说,从光盘中读取上述信息,且将其提供给解码器,并将其临时存储在解码缓冲器中。然后,在指定解码时间从解码缓冲器中读出每一个信息项,并对其进行解码。然后,将所解码的信息项存储在输出缓冲器中。然后,在指定输出时间将已解码并存储在输出缓冲器中的信息输出。

[0014] 假定对字幕图像数据和菜单屏幕显示数据中的每一个提供解码器。

发明内容

[0015] 如上所述,当从光盘上读取形成字幕数据或菜单屏幕数据的每一个信息项,并将其提供给解码器时,将其存储在解码缓冲器中,直到到达所指定的解码时间。因此,如果从光盘中读出信息项并将其提供给解码器缓冲器的时间,和从解码缓冲器中读出解码缓冲器中存储的信息项的时间,即,解码时间,之间的适当的关系受到干扰,则将大于解码器缓冲器容量的信息量,提供给解码器缓冲器。这可以导致解码器中处理的失败。

[0016] 这样,希望提供一种允许从记录介质中合适地读取形成字幕图像数据或菜单屏幕信息数据的信息项的数据结构,将数据结构记录在其上的记录介质,用于产生数据结构的创作设备、方法和程序,用于将数据结构记录在记录介质上的记录设备、方法和程序,用于验证所产生的数据是否满足某些条件的验证设备、方法和程序,以及用于制造该记录介质的制造设备和方法。

[0017] 根据本发明的一个实施例,提供包括内容数据和基于多个第一单元形成的伴随数据的数据结构,该伴随数据用于显示伴随该内容数据的信息。在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0018] 利用该配置,即使在为用于对第一单元进行解码的解码器提供的缓冲器中能够存储的段的数目局限于 N,也并不干扰解码处理。

[0019] 根据本发明的另一个实施例,提供一种记录介质,该记录介质包括记录在记录介质上的数据结构,该数据结构包括内容数据和基于多个第一单元形成的伴随数据,该伴随数据用于显示伴随该内容数据的信息。在数据结构中,在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0020] 利用该配置,当回放设备中回放该记录介质时,即使在为用于对第一单元进行解

码的解码器提供的缓冲器中能够存储的段的数目局限于 N,也并不干扰解码处理。

[0021] 根据本发明的另一个实施例,提供一种用于产生包括内容数据和基于多个第一单元形成的伴随数据的数据结构的创作设备,该伴随数据用于显示伴随该内容数据的信息。该创作设备包括产生内容数据和伴随数据的数据产生器,对数据产生器所产生的内容数据和伴随数据进行编码的编码器,以及对编码器所编码的内容数据和伴随数据进行多路复用的多路复用器。产生所述数据结构,使得在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0022] 根据本发明的另一个实施例,提供一种用于产生包括内容数据和基于多个第一单元形成的伴随数据的数据结构的创作方法,该伴随数据用于显示伴随该内容数据的信息。该创作方法包括步骤:产生内容数据和伴随数据,对所产生内容数据和伴随数据进行编码,以及对所编码的内容数据和伴随数据进行多路复用。产生所述数据结构,使得在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0023] 根据本发明的另一个实施例,提供一种允许计算机执行用于产生数据结构的创作方法的创作程序,该数据结构包括内容数据和基于多个第一单元形成的伴随数据,该伴随数据用于显示伴随该内容数据的信息。该创作方法包括步骤:产生内容数据和伴随数据,对所产生内容数据和伴随数据进行编码,以及对所编码后的内容数据和伴随数据进行多路复用。产生所述数据结构,使得在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0024] 利用该配置,在回放所产生数据的回放设备中,即使在为用于对第一单元进行解码的解码器提供的缓冲器中能够存储的段的数目局限于 N,也并不干扰 解码处理。

[0025] 根据本发明的另一个实施例,提供一种记录设备,用于在记录介质上记录包括内容数据和基于多个第一单元形成的伴随数据的数据结构,该伴随数据用于显示伴随该内容数据的信息。该记录设备包括产生内容数据和伴随数据的数据产生器,对数据产生器所产生的内容数据和伴随数据进行编码的编码器,对编码器编码后的内容数据和伴随数据进行多路复用的多路复用器,转换器,用于将多路复用器多路复用后的内容数据和伴随数据转换为适合在记录介质上记录的格式,以及记录器,用于将转换器转换后的内容数据和伴随数据记录在记录介质上。该记录器在记录介质上记录的数据结构,使得在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0026] 根据本发明的另一个实施例,提供一种记录方法,用于在记录介质上记录包括内容数据和基于多个第一单元形成的伴随数据的数据结构,该伴随数据用于显示伴随该内容数据的信息。该记录方法包括步骤:产生内容数据和伴随数据,对所产生的内容数据和伴随

数据进行编码,对编码后的内容数据和伴随数据进行多路复用,将多路复用后的内容数据和伴随数据转换为适合在记录介质上记录的格式,以及将转换后的内容数据和伴随数据记录在记录介质上。将数据结构记录在记录介质上,使得在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0027] 根据本发明的另一个实施例,提供一种记录程序,其允许计算初执行将数据结构记录在记录介质上的记录方法,该数据结构包括内容数据和基于多个第一单元形成的伴随数据,该伴随数据用于显示伴随该内容数据的信息。该记录方法包括步骤:产生内容数据和伴随数据,对所产生的内容数据和伴随数据进行编码,对编码后的内容数据和伴随数据进行多路复用,将多路复用后的内容数据和伴随数据转换为适合在记录介质上记录的格式,以及将转换后的内容数据和伴随数据记录在记录介质上。将数据结构记录在记录介质上,使得在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0028] 利用该配置,当在回放设备中回放记录设备所记录的记录介质时,即使在为用于对第一单元进行解码的解码器提供的缓冲器中能够存储的段的数目局限于 N,也并不干扰解码处理。

[0029] 根据本发明的另一个实施例,提供一种验证设备,用于执行对数据结构的验证,该数据结构包括内容数据和基于多个第一单元形成的伴随数据,该伴随数据用于显示伴随该内容数据的信息。在该数据结构中,在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中存储的第一单元的数目的上限,且输入具有该数据结构的数据。该验证设备包括确定单元,用于确定是否在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0030] 根据本发明的另一个实施例,提供一种验证方法,用于执行对数据结构的验证,该数据结构包括内容数据和基于多个第一单元形成的伴随数据,该伴随数据用于显示伴随该内容数据的信息。在该数据结构中,在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中存储的第一单元的数目的上限。输入具有该数据结构的数据。该验证方法包括步骤:确定是否在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0031] 根据本发明的另一个实施例,提供一种验证程序,允许计算机执行用于对数据结构进行验证的验证方法,该数据结构包括内容数据和基于多个第一单元形成的伴随数据,

该伴随数据用于显示伴随该内容数据的信息。在该数据结构中,在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。输入具有该数据结构的数据。该验证方法包括步骤:确定是否在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0032] 利用该配置,能够预先知道,在回放设备中回放包括内容数据和伴随数据的数据结构时,如果为用于对第一单元进行解码的解码器提供的缓冲器中能够存储的段的数目局限于 N,则解码处理将被干扰。

[0033] 根据本发明的另一个实施例,提供一种记录介质的制造设备,在记录介质上记录包括内容数据和基于多个第一单元形成的伴随数据的数据结构,该伴随数据用于显示伴随该内容数据的信息。该制造设备包括产生内容数据和伴随数据的数据产生器,对数据产生器所产生的内容数据和伴随数据进行编码的编码器,对编码器编码后的内容数据和伴随数据进行多路复用的多路复用器,转换器,用于将多路复用器多路复用后的内容数据和伴随数据转换为适合在记录介质上记录的格式,母盘制作单元,用于制作记录包括转换器转换后的内容和伴随数据的数据结构的记录介质的母盘,以及记录介质再现单元,用于基于母盘制作单元所制作的母盘再现记录介质。在数据结构中,基于编码器所执行的编码和多路复用器所执行的多路复用,在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0034] 根据本发明的另一个实施例,提供一种记录介质的制造方法,在记录介质上记录包括内容数据和基于多个第一单元形成的伴随数据的数据结构,该伴随数据用于显示伴随该内容数据的信息。该制造方法包括步骤:产生内容数据和伴随数据,对所产生的内容数据和伴随数据进行编码,对编码后的内容数据和伴随数据进行多路复用,将多路复用后的内容数据和伴随数据转换为适合在记录介质上记录的格式,制作记录包括转换后的内容和伴随数据的数据结构的记录介质的母盘,以及用于基于所制作的母盘再现记录介质。在数据结构中,基于编码和多路复用,在定义时间信息的多个第一单元中的目标第一单元基于第一单元顺序的时间被设置得晚于在排列得早于所述目标第一单元 N 个第一单元量的第一单元中定义的解码时间,其中 N 表示为解码第一单元的解码器所提供的缓冲器中可存储的第一单元的数目的上限。

[0035] 利用该配置,当回放设备中回放所制造母盘光盘所再现的记录介质时,即使在为用于对第一单元进行解码的解码器提供的缓冲器中能够存储的段的数目局限于 N,也并不干扰解码处理。

[0036] 根据本发明的另一个实施例,提供一种数据结构,包括内容数据以及基于多个段形成的伴随数据,该伴随数据用于显示伴随该内容数据的信息。所述多个段的每一个是包括关于伴随数据结构的结构信息的段、包括关于用于显示伴随数据的窗口画面的窗口画面

信息的段、包括关于伴随数据颜色的颜色信息的段、包括关于伴随数据的图像的图像信息的段、以及包括关于伴随数据的分割点的分割点信息的段之一。从包括结构信息的至少一个项目和分割点信息的一个项目的所述多个段中形成显示集。从含有包括窗口画面信息的一个或多个段和包括颜色信息的一个或多个段的一个或所述多个显示集中形成时间点。伴随数据是一组时间点。将存储结构信息、窗口画面信息、颜色信息、或图像信息的目标段的解码时间设置为晚于排列得早于所述目标段 N 个段的量的段的解码时间。

[0037] 利用该配置,即使在为用于对第一单元进行解码的解码器提供的缓冲器中能够存储的段的数目局限于 N,也并不干扰解码处理。

附图说明

[0038] 图 1 示意性的描述了 BD-ROM 数据模型;

[0039] 图 2 示意性的描述了索引表;

[0040] 图 3 描述了表示剪辑 AV 流,剪辑信息,剪辑,播放项和播放列表之间关系的统一建模语言 (UML);

[0041] 图 4 描述了由多个播放列表引用同一个剪辑的方法;

[0042] 图 5 示意性的描述了一个子路径;

[0043] 图 6 示意性的描述了记录在记录介质上的文件的管理结构;

[0044] 图 7A 和 7B 示意性的描述了 BD 虚拟播放器所执行的操作;

[0045] 图 8 示意性的描述了 BD 虚拟播放器所执行的操作;

[0046] 图 9 示意性的描述了本发明的一个实施例中,图像显示系统所使用的面板结构的实例;

[0047] 图 10 示意性的描述了活动图像面板,字幕面板和图形面板的分辨率和显示颜色的实例;

[0048] 图 11 是描述组合活动图像面板,字幕面板和图形面板的图形处理器的配置的模块图;

[0049] 图 12 示意性的描述了调色板的输入 / 输出数据的实例;

[0050] 图 13 示意性的描述了调色板中所存储的调色板表格的实例;

[0051] 图 14 示意性的描述了演示图形 (PG) 流;

[0052] 图 15A 和 15B 描述了显示集的配置;

[0053] 图 16 示意性的描述了 PG 流的显示集的逻辑结构;

[0054] 图 17 示意性的描述了 IG 流的显示集的逻辑结构;

[0055] 图 18 示意性的描述了显示集的存储格式;

[0056] 图 19 示意性的描述了表示演示合成段 (PCS) 的结构实例的语法;

[0057] 图 20 示意性的描述了表示模块 `composition_object()` 的结构实例的语法;

[0058] 图 21 示意性的描述了表示窗口定义段 (WDS) 的结构实例的语法;

[0059] 图 22 示意性的描述了表示模块 `window()` 的结构实例的语法;

[0060] 图 23 示意性的描述了菜单屏幕和按钮的配置;

[0061] 图 24 示意性的描述了表示交互合成段 (ICS) 的头部信息的结构实例的语法;

[0062] 图 25 示意性的描述了表示模块 `interactive_composition_data_fragment()` 的

结构实例的语法；

[0063] 图 26 示意性的描述了表示模块 page() 的结构实例的语法；

[0064] 图 27 示意性的描述了表示模块 button_overlap_group() 的结构实例的语法；

[0065] 图 28 示意性的描述了表示模块 button() 的结构实例的语法；

[0066] 图 29 是根据本发明的一个实施例,描述 PG/IG 数据解码器模型的配置的 实例的模块图；

[0067] 图 30 示意性的描述了 PTS/DTS 缓冲器中所存储数据的配置的实例；

[0068] 图 31 是描述 PG 解码器以显示集为单元执行处理的实例的流程图；

[0069] 图 32 是描述 IG 解码器以显示集为单元执行处理的实例的流程图；

[0070] 图 33 是根据本发明的一个实施例,描述回放设备的配置的模块图；

[0071] 图 34A 和 34B 示意性的分别描述了在一个时间点内, PG 数据和 IG 数据的段的数目上限的实例；

[0072] 图 35 示意性的描述了根据本发明的一个实施例, PG 数据和 IG 数据的数据结构；

[0073] 图 36A 和 36B 示意性的描述了从缓冲器提供给 PG 解码器或 IG 解码器的包流的流动；

[0074] 图 37 示意性的描述了从缓冲器输出的 TS 包流和基于 TS 包的段的构建；

[0075] 图 38 是描述用于制作将包括视频数据, 音频数据, PG 数据和 IG 数据的多路复用流记录于其上的记录介质的处理的流程图；

[0076] 图 39 是特别描述对数据流执行的验证处理的流程图；

[0077] 图 40 是描述执行创作处理的创作设备的配置的实例的模块图；

[0078] 图 41 示意性的描述了创作设备中使用的用户界面屏幕；

[0079] 图 42 是描述执行创作处理的创作设备的配置的另一个实例的模块图；以及

[0080] 图 43 示意性的描述了通过回放设备进行光盘再现的制造方法。

具体实施方式

[0081] 以下将结合附图,对本发明的实施例进行说明。为了便于理解,首先描述存储在 BD-ROM 中的内容,即, AV 数据,的管理结构,其中 BD-ROM 为“蓝光光盘只读格式第 1.0 版第 3 部分影音规范”所定义的只读蓝光光盘。这里, BD-ROM 中的管理结构被称之为“BDMV 格式”。

[0082] 通过 MPEG 视频或 MPEG 音频编码方法编码、根据 MPEG2 系统多路复用的比特流,被称之为“剪辑 AV 流”(或简称 AV 流)。通过关于蓝光光盘的标准之一的“蓝光光盘只读格式第 2 部分”中定义的文件系统,将剪辑 AV 流作为文件记录在光盘上。上述文件被称之为“剪辑 AV 流文件(或简称 AV 流文件)”。

[0083] 剪辑 AV 流文件从文件系统的角度来说管理单元,并且它们对于用户并不一定很容易理解。为了增强用户的方便性,应提供用于将多个分离的剪辑 AV 流文件组合成一段电影内容,并将其回放的机制,或只对一些剪辑 AV 流文件进行回放的机制。这也需要在光盘上将用于便于专门回放或任意存取回放操作的信息作为数据库进行记录。通过关于蓝光光盘的标准之一的“蓝光光盘只读格式第 3 部分”定义该数据库。

[0084] 附图 1 示意性的描述了 BD-ROM 的数据模型。如附图 1 所示, BD-ROM 的数据结构包

括 4 层。最底层是放置剪辑 AV 流的层（为了方便起见，这一层被称之为“剪辑层”）。第 2 最底层的是放置指定剪辑 AV 流回放部分的电影播放列表和播放项的层（为了方便起见，这一层被称之为“播放列表层”）。比第 2 最底层高的层，是放置包括指定电影播放列表回放次序的命令的电影对象的层（为了方便起见，这一层被称之为“对象层”）。最顶层是放置管理该 BD-ROM 中存储的标题等的索引表的层（为了方便起见，这一层被称之为“索引层”）。

[0085] 下面将给出剪辑层的详细描述。剪辑 AV 流是以 MPEG2 TS 的形式多路复用视频数据和音频数据的比特流。将关于剪辑 AV 流的信息作为剪辑信息记录在文件中。

[0086] 在剪辑 AV 流中，还多路复用用于显示字幕的演示图形 (PG) 流或用于显示菜单的交互图形 (IG) 流。

[0087] 剪辑 AV 流文件和存储相关剪辑信息的剪辑信息文件被认为是一个独立的对象，被称之为“剪辑”。也就是说，剪辑是一个由剪辑 AV 流和剪辑信息构成的独立的对象。

[0088] 通常，可以将一个文件作为一个字节串来对待。剪辑 AV 流文件中的内容，在时间轴上扩张，并且主要指定剪辑中的入口点作为时间基数。如果提供一个预定剪辑存取点的时间标记，则可以将相应当剪辑信息文件用于寻找关于在相关剪辑 AV 流文件中开始读取数据的地址的地址信息。

[0089] 下面将给出播放列表层的详细描述。电影播放列表，由指定所要回放的 AV 流文件的一组回放起始点 (IN 点) 和回放结束点 (OUT 点) 构成，并且还指明了特定 AV 流文件的回放部分。关于一个回放起始点和一个回放结束点的信息，被称之为“播放项”(PlayItem)。一个电影播放列表由一组播放项构成。回放一个播放项，是回放由播放项所引用的 AV 流文件的回放部分。也就是说，在关于播放项中 IN 点和 OUT 点的信息的基础上，回放剪辑中的对应区域。

[0090] 下面将给出对象层的详细描述。一个电影对象包括用于将 HDMV 导航命令程序 (HDM 程序) 与电影对象相关联的终端信息。HDMV 程序是用于控制播放列表的回放的命令。终端信息包括用于允许用户在 BD-ROM 播放器上执行交互操作的信息。基于该终端信息，可以控制诸如显示菜单屏幕或标题搜索之类的用户操作。

[0091] BD-J 对象是 Java™ 程序对象。BD-J 对象并不直接涉及本发明，因此省略对其的详细解释。

[0092] 下面将给出索引层的详细描述。索引层包括一个索引表。该索引表是定义 BD-ROM 光盘标题的顶层表格。基于索引表中存储的标题信息，在 BD-ROM 驻留系统软件中，通过模块管理器控制 BD-ROM 光盘的回放。

[0093] 更具体的说，如附图 2 中示意性示出的，索引表中的任意条目被称之为“标题”，且索引表中所输入的第一回放，顶层菜单以及标题 #1, #2 等都是标题。每一个标题表示到电影对象或 BD-J 对象的链接。每一个标题表示 HDMV 标题或 BD-J 标题之一。

[0094] 现在，可以假定 BD-ROM 光盘中存储的内容是电影。然后，第一回放是在电影之前播放的电影公司的商业视频（电影预告片）。顶层菜单是用于选择电影回放，章节搜索，字母或语言的设置，特定视频的回放等的菜单屏幕。标题是从顶层菜单中选择的视频。标题可能被配置为一个菜单屏幕。

[0095] 附图 3 是描述表示上述剪辑 AV 流，剪辑信息（流属性），剪辑，播放项，以及播放列表中关系的统一建模语言 (UML)。播放列表与一个或多个播放项相关，且该播放项与一个剪

辑相关。具有不同起始点和结束点的多个播放项,可以与一个剪辑相关。一个剪辑 AV 流文件由一个剪辑引用。同样,一个剪辑信息文件也由一个剪辑引用。剪辑 AV 流文件和剪辑信息文件具有一对一的对应性。通过上述结构的定义,无需改变剪辑流文件,就可以定义用于仅回放任意部分的无损回放次序。

[0096] 如附图 4 所示,多个播放列表能够引用同一个剪辑。相反,一个播放列表能够指定多个剪辑。播放列表中播放项所指出的 IN 点和 OUT 点引用一个剪辑。在附图 4 示出的实例中,播放列表 310 的播放项 320 引用剪辑 300,且播放列表 311 的播放项 321 的 IN 点和 OUT 点引用剪辑 300 的一个区域。播放列表 311 的播放项 322 的 IN 点和 OUT 点引用剪辑 301 的一个区域,且此外,播放列表 312 的播放项 323 的 IN 点和 OUT 点引用剪辑 301 的另一个区域。

[0097] 如附图 5 中示出的一个实例所指出的,播放列表包括与所要回放的播放项相关的主路径,以及与子播放项相关的子路径。例如,子路径可以是添加至该播放项的后记录(配音)播放项。只有当播放列表满足预定条件时,才提供子播放项,虽然这里省略了细节的描述。

[0098] 下面将结合附图 6,论述“蓝光光盘只读格式第 3 部分”定义的 BD-ROM 上记录的文件的管理结构。通过目录结构,对文件进行分层管理。首先,在记录介质上创建一个目录(附图 6 示出实例中的根目录)。通过一个记录/回放系统管理该目录之下的目录。

[0099] 根目录之下,设置目录“BDMV”和目录“CERTIFICATE”。目录“CERTIFICATE”中存储关于版权的信息。目录“BDMV”中,存储根据附图 1 所论述的数据结构。

[0100] 直接在目录“BDMV”之下,设置两个文件,即文件“index.bdmv”和文件“MovieObject.bdmv”。在目录“BDMV”之下,设置目录“PLAYLIST”,目录“CLIPINF”,目录“STREAM”,目录“AUXDATA”,目录“META”,目录“BDJO”,目录“JAR”以及目录“BACKUP”。

[0101] 在文件“index.bdmv”中,指出目录“BDMV”的内容。也就是说,最顶层的文件“index.bdmv”对应于索引层中的索引表。在文件“MovieObject.bdmv”中,存储关于至少一个电影对象的信息。也就是说,文件“MovieObject.bdmv”对应于上述对象层。

[0102] 目录“PLAYLIST”是存储播放列表的数据库的目录。也就是说,目录“PLAYLIST”包括关于电影播放列表的文件“xxxxx.mpls”。每一个文件“xxxxx.mpls”为相应的电影播放列表而创建。在文件名中,“.”(句号)之前的“xxxxx”是五个数字,且句号之后的“mpls”是只有该类型文件才有的固定扩展名。

[0103] 目录“CLIPINF”是放置剪辑数据库的目录。也就是说目录“CLIPINF”包括作为对应剪辑 AV 流文件的剪辑信息文件的文件“zzzzz.clpi”。在文件名中,“.”(句号)之前的“zzzzz”是五个数字,且句号之后的“clpi”是只有该类型文件才有的固定扩展名。

[0104] 目录“STREAM”是放置作为真实数据的 AV 流文件的目录。也就是说,目录“STREAM”包括与对应剪辑信息文件相关的剪辑 AV 流文件。剪辑 AV 流文件是 MPEG2 传输流(在下文中称之为“MPEG2 TSs”),且其文件名为“zzzzz.m2ts”。在文件名中,句号之前的“zzzzz”被设置为与对应剪辑信息文件文件名中句号之前的部分相同。这将促使剪辑信息文件和剪辑 AV 流文件之间的关系更易于理解。

[0105] 目录“AUXDATA”是放置用于显示菜单的声音文件,字体文件,字体索引文件,位图文件等的目录。在文件“sound.bdmv”中,存储与 HDMV 交互图形流应用程序相关的声音数

据。文件名固定为“sound.bdmv”。在文件“aaaaa.otf”中,存储用于字幕显示或 BD-J 应用程序的字体数据。在文件名中,句号之前的“aaaaa”是五个数字,且句号之后的“otf”是只有该类型文件才有的固定扩展名。文件“bdmv.fontindex”是字体索引文件。

[0106] 在目录“META”中,存储元数据文件。在目录“BDJO”和目录“JAR”中,存储与上述 BD-J 对象相关的文件。在目录“BACKUP”中,存储上述目录和文件的备份数据。目录“META”,目录“BDJO”,目录“JAR”和目录“BACKUP”并不直接涉及本发明,因此省略对其的解释。

[0107] 当播放器中放置具有上述数据结构的光盘时,首先,需要播放器将从光盘读取的电影对象中指出的命令,转换为用于控制播放器硬件的统一命令。在内置 ROM 中存储用于执行上述命令转换的软件。根据 BD-ROM 标准,该软件担任光盘和播放器之间的媒介,从而允许播放器执行操作,且因此,该软件被称之为“BD 虚拟播放器”。

[0108] 附图 7A 和 7B 示意性的描述了 BD 虚拟播放器执行的操作。附图 7A 描述了当置入光盘时,BD 虚拟播放器所执行的操作的实例。在步骤 S30 中,在播放器中放置光盘,并对光盘进行初始化。然后,在步骤 S31 中,初始化存储光盘中使用的通用参数的寄存器。然后,在步骤 S32 中,执行从光盘读取的程序。应该注意的是,例如,当光盘第一次置入播放器中时,初始化存取是为了对光盘进行首次回放。

[0109] 附图 7B 描述了用户通过按压,例如,播放键,给出回放光盘的指令时,BD 虚拟播放器所执行操作的实例。在步骤 S40 中,播放器处于暂停状态,然后,用户利用,例如,远程控制器给出指令(在附图 7B 中,该操作由 UOP(用户操作)表示)。然后,在步骤 S41 中,初始化存储通用参数的寄存器。这样,重置寄存器将不是所必须的。在步骤 S42 中,播放器回放播放列表(回放阶段)。

[0110] 下面将结合附图 8,论述通过运行电影对象的播放列表的回放。假定通过用户操作给出回放标题号码 #1 的内容的指令。为了响应回放该内容的指令,播放器提交附图 2 中示出的索引表,从而获得对应于标题号码 #1 的对象号码。如果对应于标题号码 #1 的对象号码为 #1,则播放器启动执行控制对象 #1。

[0111] 在附图 8 示出的实例中,电影对象 #1 中的程序包括 2 条线路,且第一条线路中的命令表示“播放播放列表(1)”。因此,播放器开始回放播放列表 #1。播放列表 #1 包括至少一个播放项,并顺序回放多个播放项。在结束播放列表 #1 中播放项的回放之后,播放器恢复电影对象 #1 的激活,并执行第二线路中的命令。在附图 8 示出的实例中,第二线路中的命令表示“跳转至顶层菜单”,且播放器执行该命令,从而启动索引表中指示的顶层菜单的电影对象。

[0112] 这里,给出适合于本发明实施例的图像显示系统的说明。该实施例中,基于如附图 9 所示的面板配置图像显示系统。在最低(向后)层上显示用于处理播放列表中指定的图像(主要是活动图像数据)的活动图像面板 10。在活动图像面板 10 的上一层,显示处理在活动图像回放过程中所要回放的字幕数据的字幕面板 11。在最顶(向前)层上,显示用于处理图形数据的交互图形面板 12,该图形数据,是诸如显示菜单屏幕的字符数据和用于按钮图像的位图数据之类的图形数据。这三个面板可以进行组合,以便在一个显示屏幕上显示。

[0113] 可以独立显示活动图像面板 10,字幕面板 11,和交互图形面板 12。它们具有如附图 10 所示出的分辨率和显示颜色。对于活动图像面板 10,分辨率为 1920(像

素)×1080(行),且一个像素的数据长度为16位。包括亮度信号Y和色差信号Cb和Cr的YCbCr系统为4:2:2。YCbCr(4:2:2)是一个颜色系统,其中,亮度信号Y和色差信号Cb和Cr均具有8位每像素,且用于两个水平像素的色差信号Cb和Cr形成一个颜色数据。对于字幕面板11和交互图形面板12,分辨率为1920(像素)×1080(行),且每像素的采样深度为8位。作为颜色系统,该系统使用了应用256色调色板的8位颜色映射地址。

[0114] 在字幕面板11和交互图形面板12中,能够实现256级的 α 混合(alphablending)。当将字幕面板11或交互图形面板12与其他面板进行组合时,不透明性可以设置为256级。可以对每一像素进行不透明性的设置。下文中,不透明性 α 可以表示为($0 \leq \alpha \leq 1$)。当不透明性 $\alpha = 0$ 时,字幕面板11或交互图形面板12是完全透明的。当不透明性 $\alpha = 1$ 时,字幕面板11或交互图形面板12是完全不透明的。

[0115] 字幕面板11处理便携式网络图形(PNG)的图像数据。交互图形面板12也能够处理PNG图像数据。PNG格式采样深度的范围从1到16位每像素。当采样深度为8位或16位时,可以添加 α 通道,也就是,关于每个像素不透明性的不透明信息(下文中也称为“ α 数据”)。当采样深度为8位时,不透明性可以指定在256级之内。通过利用不透明性信息(α 通道)执行 α 混合。可以使用256色调色板图像,且可以提高索引数目,指定预先提供的调色板的颜色数目(索引)。

[0116] 字幕面板11或交互图形面板12所处理的图像数据并不限制为PNG格式。也可以处理其他压缩编码方法所压缩的图像数据,诸如联合图像专家组(JPEG),游程长度压缩图像数据,未压缩位图图像数据。

[0117] 附图11描述了根据附图9所示显示系统和附图10所示分辨率和显示颜色,合并三个面板的图形处理器的实例的配置。将活动图像面板10的活动图像数据提供给422/444转换电路20。在422/444转换电路20中,将活动图像数据的颜色系统从YCbCr(4:2:2)转换为YCbCr(4:4:4),且然后,将转换后的数据输入至乘法器21。

[0118] 将字幕面板11的图像数据输入至调色板22A,并作为RGB(4:4:4)图像数据输出。如果根据 α 混合指定图像数据的不透明性,则从调色板22A输出所指定的不透明性 α 1($0 \leq \alpha \leq 1$)。

[0119] 附图12描述了调色板22A的输入/输出数据的一个实例。在调色板22A中,将对应于诸如PNG文件的调色板信息存储为一个表格。调色板22A涉及通过以输入的8位像素数据作为地址的索引号码。基于该索引号码,调色板22A输出RGB(4:4:4)数据,每个颜色具有8位。同时,也从调色板22A中输出表示不透明性的 α 通道数据 α 。

[0120] 附图13描述了调色板22A中所存储的调色板表格的实例。三个主要颜色R,G,B,每一个都具有8位,且将不透明性 α 分配至从[0x00]到[0xFF]的每一个256色索引值([0x]表示16进制符号)。调色板22A涉及以所输入PNG图像数据为基础的调色板表格,并为每一个像素输出均为8位的RGB数据,以及对应于图像数据所指定索引值的不透明性 α 。

[0121] 将从调色板22A输出的RGB数据提供给RGB/YCbCr转换电路22B。在RGB/YCbCr转换电路22B中,将RGB数据转换为由亮度信号Y以及色差信号Cb和Cr所表示的数据(上述数据共称为“YCbCr数据”),每个信号的数据长度为8位。进行该转换的原因是为了利用通用数据格式,也就是作为活动图像数据的数据格式的YCbCr数据,对面板进行合并。

[0122] 将 YCbCr 数据以及从 RGB/YCbCr 转换电路 22B 中输出的不透明性数据 $\alpha 1$, 输入至乘法器 23。在乘法器 23 中, 将所输入的 YCbCr 数据与不透明性数据 $\alpha 1$ 相乘。将相乘的结果输入至加法器 24 的一个输入端。在乘法器 23 中, 亮度信号 Y 以及色差信号 Cb 和 Cr 的每一个, 与不透明性数据 $\alpha 1$ 相乘。将不透明性数据 $\alpha 1$ 的补数 $(1-\alpha 1)$ 提供给乘法器 21。

[0123] 在乘法器 21 中, 从 422/444 转换电路 20 输入的活动图像数据, 与不透明性数据 $\alpha 1$ 的补数 $(1-\alpha 1)$ 相乘。将相乘的结果输入至加法器 24 的另一个输入端。在加法器 24 中, 乘法器 21 和 23 的相乘结果进行相加。然后, 合并活动图像面板 10 和字幕平面 11。将加法器 24 相乘的结果输入至乘法器 25。

[0124] 将交互图形面板 12 的图像数据输入至调色板 26A, 并作为 RGB(4:4:4) 图像数据输出。如果利用 α 混合为图像数据指定了不透明性, 则从调色板 26A 输出所指定的不透明性 $\alpha 2 (0 \leq \alpha 2 \leq 1)$ 。将从调色板 26A 输出的 RGB 数据提供给 RGB/YCbCr 转换电路 26B, 并转换为 YCbCr 数据, 该数据将统一为活动图像数据的数据格式。将从 RGB/YCbCr 转换电路 26B 输出的 YCbCr 数据输入至乘法器 27。

[0125] 如果交互图形面板 12 的图像数据为 PNG 格式, 则为每个像素设置不透明性 $\alpha 2 (0 \leq \alpha 2 \leq 1)$ 。将不透明性数据 $\alpha 2$ 提供给乘法器 27。在乘法器 27 中, 将从 RGB/YCbCr 转换电路 26B 输入的 YCbCr 数据的亮度信号 Y 以及色差信号 Cb 和 Cr 的每一个, 与不透明性数据 $\alpha 2$ 相乘。乘法器 27 中相乘的结果输入至加法器 28 的一个输入端。将不透明性数据 $\alpha 2$ 的补数 $(1-\alpha 2)$ 提供给乘法器 25。

[0126] 在乘法器 25 中, 加法器 24 中相加的结果与不透明性数据 $\alpha 2$ 的补数 $(1-\alpha 2)$ 相乘。将乘法器 25 中相乘的结果输入至加法器 28 的另一个输入端, 且与乘法器 27 中相乘的结果相加。然后, 交互图形面板 12 与活动图像面板 10 和字幕面板 11 的合成图像相合并。

[0127] 在字幕面板 11 或交互图形面板 12 中, 没有图像的区域的不透明性数据 α 被设置为 0, 以便能够透过面板 11 或 12 看到字幕面板 11 或交互图形面板 12 下方所显示的面板。例如, 在活动图像面板 11 上显示的活动图像数据能够作为字幕面板 11 或交互图形面板 12 的背景进行显示。

[0128] 通过硬件或软件, 均可实现附图 11 中示出的图形处理器的配置。

[0129] 结合附图 14, 给出了演示图形 (PG) 流和交互图形 (IG) 流的说明。如上所述, PG 流是包括用于显示字幕的字幕图像数据的流。更具体地说, PG 流包括形成位图数据的字幕图像数据本身和表示如何显示该字幕图像数据的数据。IG 流是包括用于显示菜单的数据的流, 且更具体地说, 用于菜单屏幕的按钮图像数据, 表示如何显示该按钮图像数据的数据, 以及该按钮所指示的命令。

[0130] PG 流或 IG 流包括在剪辑 AV 流中, 且形成在包括一个或多个显示集的多个时间点单元内。该显示集和时间点可以认为是分别在 MPEG2 中定义的一个图像和图组 (GOP)。

[0131] 也就是说, 显示集是用于显示字幕的最小单元, 且对应于一个字幕图像。一个时间点, 是由一个或多个显示集所形成的。通常, 时间点是由显示项单元所形成的。也就是说, 一个时间点是由某一显示项回放过程中, 用于显示所要显示的字幕的显示集而形成的。用于解码 PG 流的 PG 解码器, 清除字幕面板 11 和多个时间点单元中的不同存储器。

[0132] 对于 PG 流, 如附图 14 所示, 一个时间点中可以包括多个显示集。相反, 对于 IG 流, 如果 IG 流是一个预先载入的 IG 流, 则一个时间点中仅包括一个显示集。

[0133] 事实上,用于形成时间点的基本单元并不局限于播放项,且形成时间点的所需项可以由创作方确定。在PG流的情况下,能够由用于显示电影中形成对话的多个字幕集的多个显示集形成一个时间点。作为选择,可以通过词语来分离该对话,且可以由用于显示相应词语的多个显示集形成一个时间点。

[0134] 附图 15A 和 15B 示意性的描述了显示集的配置。如附图 15A 所示,PG 流的显示集包括 5 段,诸如演示合成段 (PCS),窗口定义段 (WDS),调色板定义段 (PDS),对象定义段 (ODS),以及结束段。虽然这里并未讨论每一段的细节,但以下仍将简要说明每一段。在 PCS 中,存储关于字幕的基本信息。WDS 定义了用于显示字幕的显示画面,该显示画面被称为“窗口”。WDS 固定在该时间点中。PDS 定义了关于字幕的颜色信息,且包括,例如,一个索引颜色信息表。ODS 包括关于字幕的配置信息。例如,通过预定压缩编码方法,诸如游程长度压缩编码,对用于显示该字幕的位图数据进行压缩编码。结束段是表示该显示集结束的段。

[0135] 如附图 15B 所示,IG 流的显示集包括 4 段,即,交互合成段 (ICS),PDS,ODS,以及结束段。在 4 段之中,ICS 是存储 IG 流的基本结构的段。PDS,ODS,和结束段与 PG 流的显示集中的类似。也就是说,PDS 是存储关于按钮图像的颜色信息的段。ODS 是存储按钮图像的的配置的信息。更具体地说,在 ODS 中,通过预定压缩编码方法,诸如游程长度压缩,对按钮图像自身,例如,用于显示按钮图像的位图数据,进行压缩编码并存储。结束段是表示该显示集结束的段。

[0136] 附图 16 示意性的描述了 PG 流的显示集的逻辑结构。4 段构成的一组,诸如 PCS,WDS,PDS,ODS,以及结束段(未示出),形成一个显示集,如附图 16 中 A 部分和 B 部分所指出的。唯一的一个 PCS 可以一直存储在一个显示集内。唯一的一个 WDS 可以几乎一直存储在一个显示集内。然而,如果下面所要讨论的,PCS 中定义的标记 `palette_up_date_flag` 表示 1,则 WDS 是可选的。PDS 和 ODS 是可选的,且在一个显示集中包括 PDS 和 ODS 并不是必需的。

[0137] 如附图 16 中 C 部分所指出的,一个时间点由一个或多个显示集形成。虽然这里没有讨论细节,但可以在 PCS 的模块 `composition_descriptor()` 中包括的域 `composition_state` 中定义时间点起始显示集。基于该时间点起始显示集,对时间点进行定义。例如,将时间点定义为一组显示集,这一组显示集从在 PCS 中对时间点起始显示集进行定义的显示集开始,到在 PCS 中在 PCS 中对下一个时间点起始显示集进行定义的显示集的前一个显示集为止。下面,将讨论时间点定义的一个具体实例。如附图 16 中 D 部分所指出的,PG 流包括一个或多个时间点。

[0138] 附图 17 示意性的示出了 IG 流的显示集的逻辑结构。如附图 17 中 A 部分所指出的,除了段的配置以外,交互图形的显示集的整体层次结构和 PG 流的结构相同。在 IG 流的显示集中,与 PG 流相同,基于 ICS 的模块 `composition_descriptor()` 中包括的域 `composition_state` 中定义的时间点起始显示集,对时间点进行定义。在 PG 流中,一个显示集中只包括一个表示结构的 PCS。相反,在 IG 流中,在一个显示集中包括多个表示结构的 ICS。也就是说,可能多个 ICS 定义的是一个单独的模块 `composition_descriptor()`。此外,PG 流或 IG 流,在一个显示集中只包括一个结束段。与 PG 流中相同,PDS 和 ODS 是可选的,且在显示集中包括 PDS 和 ODS 并不是必需的。对于 IG 流的显示集,为了简单的描述,省略了更详细的说明。

[0139] 附图 18 描述了上述显示集的实例的存储格式。如上所述,用于显示字幕的显示集和用于显示按钮图像的显示集,分别与诸如 PG 流和 IG 流的剪辑 AV 流进行多路复用。如附图 18 中的 A 部分所指出的,该显示集由多个段形成。

[0140] 如附图 18 中的 B 部分所指出的,形成该显示集的多个段彼此通过头信息进行区分,且多个段均存储在打包基本码流 (PES) 包的有效负载中。每一个 PES 包的最大尺寸固定为 64k 字节 (KB)。如果一个具有相对较大尺寸的 ODS 或 ICS,超过了最大尺寸,则将被划分为较小的部分,且然后,将所划分的部分填入多个 PES 包的有效负载。另一方面,由于大多数情况下 PCS, WDS, 和 PDS 都小于 64KB, 所以一个段可以存储在一个 PES 包中。在每一个 PES 包中,在 PES 的头部,存储表示有效负载,即, PCS, WDS, PDS, 或 ODS, 中 (在 PG 流的情况下) 存储的数据的类型的信息,或表示包顺序的识别信息。

[0141] 在 PES 的头部,还存储了指定该 PES 包解码时间的 DTS (解码时间标记) 和指定所解码数据的输出时间的演示时间标记 (PTS)。也就是说,当系统时间时钟 (STC) 与 DTS 相一致时,开始对 PES 包有效负载中存储的数据进行解码,该系统时间时钟,是在 MPEG 系统所定义的基准解码器中所设置的。当 STC 与 PTS 相一致时,输出所解码的数据。

[0142] PTS 和 DTS 是可选的,且如果预定存取单元的第一段部分包括在 PES 包的有效负载中,则将 PTS 和 DTS 添加至 PES 的头部。例如,如果划分一个段,并将其存储至多个 PES 包中,则在存储第一段部分的 PES 包中仅存储 PTS 和 DTS。

[0143] 如附图 18 的 C 部分所指出的,进一步将每一个 PES 包划分为更小的部分,并存储在形成 MPEG TS 的传输流 (TS) 包中。在包识别码 (PID) 中,存储 TS 包的顺序和用于识别每一个 TS 包中存储的数据的识别信息。

[0144] 以下,将具体给出直接涉及本发明的段的细节。首先,讨论 PG 流的显示集中所包括的 PCS 和 WDS。附图 19 描述了表示 PCS 结构的实例的语法。附图 19 中示出的语法是以 C 语言符号表示的,C 语言是计算机程序描述语言中的一种。其他附图中示出的语法也是以 C 语言符号表示的。

[0145] 模块 `segment_descriptor()` 包括域 `segment_type` 和域 `segment_length` (未示出)。域 `segment_type` 指定了段的类型。如果域 `segment_type` 的值为 0x16,这意味着该段是 PCS。0x 表示 16 进制符号。域 `segment_length` 表示从 `segment_length` 开始的该段的长度。

[0146] 模块 `video_descriptor()` 包括域 `video_width`, 域 `video_height`, 和域 `frame_rate` (未示出)。域 `video_width` 和域 `video_height` 表示与由该 PCS 中包括的显示集所要显示的字幕同时的,所要回放的活动图像的画面。域 `frame_rate` 指定了活动图像的画面频。

[0147] 模块 `composition_descriptor()` 包括域 `composition_number` 和域 `composition_state` (未示出)。域 `composition_number` 表示与该 PCS 相关的时间点中该段的顺序。域 `composition_state` 表示该 PCS 中包括的显示集的类型。

[0148] 以下,将简要讨论显示集的类型。存在 4 种类型的显示集,即,时间点起始显示集,时间点持续显示集,采集点显示集,以及正常状态显示集。

[0149] 时间点起始显示集是时间点的第一个显示集。在时间点与多于一个播放项相关联时,使用时间点持续显示集。例如,如果时间点与播放项 #1 和下一个播放项 #2 相关联,则

在播放项 #2 的头部设置时间点持续显示集。

[0150] 采集点显示集与字幕的解码相关,且表示安全解码的起始点。也就是说,当播放器进行任意存取时,非常可能存取的目的地是时间点的中点。然后,在任意存取之后,播放器重新开始解码,从而在采集点显示集的位置显示字幕。

[0151] 正常状态显示集与字幕的解码相关,且表示不安全解码的起始点。如果当前另一显示集所显示字幕的颜色或配置将改变,则在正常状态显示集中设置表示该改变的显示集。如果在解码和字幕回放的过程中播放器寻找正常状态显示集,则其将根据该改变,变化当前所显示字幕的颜色或配置。

[0152] 返回对 PCS 的语法的说明,域 `palette_update_flag` 是表示调色板是否更新的标记。域 `palette_id_ref` 表示调色板的 ID,该调色板涉及该显示集所显示的字幕的一个画面。也就是说,通过域 `palette_id_ref` 引用 PG 流中 PDS 的颜色信息。域 `number_of_composition_object` 表示包括该显示集所显示的字幕的一个画面中所包括的对象的数目。对域 `number_of_composition_object` 所表示的相同数目,重复下一个 FOR 语句所表示的循环。通过模块 `composition_object()`,定义该显示集所显示的每一个对象。

[0153] 附图 20 描述了表示模块 `composition_object()` 的实例的结构的语法。域 `object_id_ref` 表示通过模块 `composition_object()`,将 ODS 用于显示字幕的基准 (reference)。域 `window_id_ref` 指定了窗口的 ID,其中,通过模块 `composition_object()`,将用于显示字幕的对象进行显示。

[0154] 域 `object_cropped_flag` 是一个标记,该标记表示是否通过模块 `composition_object()`,将用于显示字幕的对象进行截取。也就是说,能够从 ODS 中截取预定尺寸的字幕。如果标记 `object_cropped_flag` 表示该对象将要被截取,基于下面所要讨论的剪辑尺寸信息,将截取部分字幕,并显示所截取的部分。

[0155] 域 `forced_on_flag` 表示是否强制显示由模块 `composition_object()` 所显示的字幕,而不管表示是否显示用户操作所给出字幕的指令。

[0156] 域 `composition_object_horizontal_position` 和 域 `composition_object_vertical_position` 分别表示屏幕上模块 `composition_object()` 所显示字幕的水平位置和垂直位置。通过对于以下所要讨论的 WDS 所表示窗口位置的相对位置,表示水平位置和垂直位置。

[0157] 如果上述域 `object_cropped_flag` 所描述的标记,表示所要截取的对象,则提供下一个 IF 语句 (`if(object_cropped_flag = 1b)`),以便指定所要截取部分的位置和尺寸。也就是说,在模块 `cropping_rectangle()` 中,分别通过域 `object_cropping_horizontal_position` 和域 `object_cropping_vertical_position`,指定所要截取部分的位置和尺寸的起点。分别通过域 `object_cropping_width` 和域 `object_cropping_height`,指定所要截取部分的宽度和高度

[0158] 附图 21 描述了表示 WDS 的实例结构的语法。模块 `segment_descriptor()` 包括域 `segment_type` 和域 `segment_length` (未示出)。域 `segment_type` 表示段的类型。如果 `segment_type` 的纸表示为 0x17,则意味着段的类型为 WDS。域 `segment_length` 表示从该域 `segment_length` 开始的这一段的长度。

[0159] 域 `number_of_windows` 表示包括该 WDS 的显示集所显示的字幕的一个画面中所包

括的窗口的数目。对域 `number_of_windows` 所表示的相同数目,重复下一个 FOR 语句所表示的循环,且通过模块 `window()`,定义每一个窗口。

[0160] 附图 22 描述了表示模块 `window()` 的实例结构的语法。域 `window_id` 指定了该窗口的 ID。域 `window_horizontal_position` 和域 `window_vertical_position` 分别表示该窗口画面的水平位置和垂直位置的开始。域 `window_width` 和域 `window_height` 分别表示该窗口的宽度和高度。

[0161] 这里,给出对 IG 流显示集中所包括 ICS 的说明。在对 ICS 说明之前,以下根据附图 23 简要讨论菜单屏幕和按钮配置。这里,先考虑附图 23 中 A 部分所指出的,显示多个按钮 300 的菜单屏幕 301。

[0162] 如附图 23 的 B 部分所指出的,以多个菜单画面层次化的构建菜单屏幕 301。如果位于最顶层的菜单画面中的某一按钮 300,借助于预定输入单元从选定状态移位到激活状态,则可以在前方最顶层显示,直接位于具有操作按钮 300 的菜单画面后部的菜单画面。下文中,借助于预定输入单元改变按钮的状态,仅称为“操作一个按钮”。

[0163] 如附图 23 的 C 部分和 D 部分所指出的,可以以多个按钮 302A,302B,302C 等层次化的构建菜单屏幕 301 上显示的一个按钮 300。换句话说,可以在单个的显示位置显示多个按钮,并选择其中一个按钮。例如,如果操作多个按钮中的一个,可以同步修改同一位置其他按钮的功能和显示。由于不需要修改菜单屏幕 301,因此是很方便的。可以在同一位置选择性显示的一组多个按钮,被称为“按钮叠加组 (BOG)”

[0164] 可以以 3 种状态提供形成 BOG 的每一个按钮,诸如正常状态,选定状态,和激活状态。如附图 23 的 E 部分所指出的,为形成 BOG 的每一个按钮,提供分别表示正常状态,选定状态,和激活状态的按钮 303A,303B,和 303C。如附图 23 的 F 部分所指出的,可以为表示 3 种状态的按钮 303A,303B,和 303C 中的每一个提供动画图像。这样,具有动画图像的按钮包括与动画图像数目相同的按钮图像数目。

[0165] 下文中,形成动画图像的多个按钮图像也称为“动画画面”。

[0166] 附图 24 描述了表示 ICS 头部信息的实例结构的语法。ICS 的头部包括模块 `segment_descriptor()`,模块 `video_descriptor()`,模块 `composition_descriptor()`,模块 `sequence_descriptor()`,以及模块 `interactive_composition_data_fragment()`。模块 `segment_descriptor()` 表示该段是一个 ICS。模块 `video_descriptor()` 表示与该菜单一起显示的画面速率和画面尺寸。模块 `composifion_descriptor()` 包括域 `composition_state`(未示出),且表示该 ICS 的状态。模块 `sequence_descriptor()` 表示该 ICS 是否与多个 PES 包相关。

[0167] 更具体地说,模块 `sequence_descriptor()` 表示当前 PES 包中包括的 ICS 是否为单个 IG 流中的第一个 ICS 或最后一个 ICS。

[0168] 如上所述,PES 包的最大数据尺寸为 64KB。如果单个 ICS 的数据尺寸超过 64KB,则 ICS 将被划分为较小的部分,且这些较小的部分将被填入 PES 包。这样,附图 24 中示出的头部只包括在含有 ICS 划分后部分的 PES 包中的第一个 PES 包和最后一个 PES 包中,且在其他 PES 包中省略该头部。如果模块 `sequence_descriptor()` 表示当前 PES 包中包括对 ICS 是第一个 ICS 和最后一个 ICS,则意味着 ICS 包括在一个单独的 PES 包中。

[0169] 附图 25 描述了表示模块 `interactive_composition_data_fragment()` 的实例

结构的语法。在附图 25 示出的实例中,所指出的模块 `interactive_composition_data_fragment()` 与模块 `interactive_composition()` 相同。域 `interactive_composition_length` 具有 24 位数据长度,且表示域 `interactive_composition_length` 之后模块 `interactive_composition()` 的长度。域 `stream_model` 具有 1 位数据长度,且表示是否对该数据流进行多路复用。也就是说,可以将 IG 流与 AV 流进行多路复用,或由 IG 流自身形成一个单独的剪辑 AV 流。

[0170] 域 `user_interface_model` 具有 1 位数据长度,且表示该数据流所显示的菜单是弹出式菜单或常规菜单(一直显示在屏幕上)。弹出式菜单的显示可以由预定输入单元的开启或关闭控制,例如,远程控制器的按钮。另一方面,借助于用户的操作,很难控制常规菜单的显示。如果域 `user_interface_model` 为 0,则该菜单为弹出式菜单。如果域 `user_interface_model` 为 1,则该菜单为常规菜单。

[0171] 如果域 `user_interface_model` 为 0,则 IF 语句,即,如果 (`stream_model = "0b"`),之后的域 `composition_time_out_pts` 和域 `selection_time_out_pts` 变为有效。域 `composition_time_out_pts` 具有 33 位数据长度,且表示所显示菜单消失的时间。域 `selection_time_out_pts` 具有 33 位数据长度,且表示在该菜单上的选择操作无效的时间。由 MPEG2 中定义的 PTS,表示域 `composition_time_out_pts` 或域 `selection_time_out_pts` 中的时间。

[0172] 域 `user_time_out_duration` 具有 24 位数据长度,且表示用于该菜单显示的自动初始化持续时间。下一个域 `number_of_pages` 具有 8 位数据长度,且表示该菜单所提供的页的数目。该域的初始值为 0。也就是说,该菜单显示具有如附图 23 中 B 部分所指出的层次结构,且如果提供多页,则域 `number_of_pages` 为 1 或更大。对与域 `number_of_pages` 所表示数目相同的数目,重复下一个 FOR 语句所表示的循环。

[0173] 附图 26 描述了表示模块 `page()` 的实例结构的语法。域 `page_id` 具有 8 位数据长度,且表示用于识别该页的 ID。域 `page_version_number` 具有 8 位数据长度,且表示该页的版本号。下一个模块 `UO_mask_table()` 指定了表示在该页的显示过程中,在输入单元上禁止执行的用户操作 (UO) 的表。

[0174] 模块 `in_effects()` 表示该页时激活的动画模块。在模块 `effect_sequence()` 中指出动画顺序。模块 `out_effects()` 表示在结束该页的显示时激活的动画模块。在模块 `effect_sequence()` 中,指出动画顺序。模块 `in_effects()` 和模块 `out_effects()` 表示当页中进行修改而发现 ICS 时所激活的动画。

[0175] 下一个域 `animation_frame_rate_code` 具有 8 位数据长度,且如果该页按钮图像作为动画示出,则表示用于设置该动画画面速率的参数。当由 V_{frm} 表示对应于该 ICS 的剪辑 AV 流文件中视频数据的画面速率,且由 A_{frm} 表示该动画画面速率时,可以由 V_{frm} 对 A_{frm} 的比率,即, V_{frm}/A_{frm} ,表示域 `animation_frame_rate_code`。

[0176] 域 `default_selected_button_id_ref` 具有 16 位数据长度,且表示显示该页时,首先选择的按钮的 ID。下一个域 `default_activated_button_id_ref` 具有 16 位数据长度,且表示到达附图 25 所讨论的域 `selection_time_out_pts` 所代表的时间时,自动激活的按钮的 ID。

[0177] 域 `palette_id_ref` 具有 8 位数据长度,且表示该页所涉及调色板的 ID。也就是

说,通过该域 `palette_id_ref` 指定 IG 流的 PDS 中的颜色信息。

[0178] 下一个域 `number_of_BOGs` 具有 8 位数据长度,且表示该页中所使用的 BOG 的数目。对与域 `number_of_BOGs` 所表示数目相同的数目,重复下一个 FOR 语句所表示的循环,然后,由模块 `button_overlap_group()` 定义每一个 BOG。

[0179] 附图 27 描述了表示模块 `button_overlap_group()` 的实例结构的语法。域 `default_valid_button_id_ref` 具有 16 位数据长度,且表示模块 `button_overlap_group()` 中定义的 BOG 所首先显示的按钮的 ID。下一个域 `number_of_buttons` 具有 8 位数据长度,且表示该 BOG 中使用的按钮的数目。对与域 `number_of_buttons` 所表示数目相同的数目,重复下一个 FOR 语句所表示的循环,然后,由模块 `button()` 定义每一个按钮。

[0180] 也就是说,如上所述,一个 BOG 可以具有多个按钮,且由模块 `button()` 定义每一个按钮的结构。该模块 `button()` 所定义的按钮结构,是所要显示的实际的按钮。

[0181] 附图 28 描述了表示模块 `button()` 的实例结构的语法。域 `button_id` 具有 16 位数据长度,且表示识别该按钮的 ID。域 `button_numeric_select_value` 具有 16 位数据长度,且表示远程控制器上分配给该按钮的数字键。域 `auto_action_flag` 具有 1 位数据长度,且表示当选择该按钮时,自动执行的分配给该按钮的功能。

[0182] 域 `button_horizontal_position` 和域 `button_vertical_position` 均具有 16 位数据长度,且指定在画面上该按钮的水平位置和垂直位置(高度)。

[0183] 模块 `neighbor_info()` 表示关于该按钮的外部信息。也就是说,模块 `neighbor_info()` 的值表示选择目标按钮的同时,操作远程控制器的指定上,下,左,或右方向的方向键时,所要选择的按钮。在模块 `neighbor_info()` 中,域 `upper_button_id_ref`,域 `lower_button_id_ref`,域 `left_button_id_ref`,以及域 `right_button_id_ref`,每一个均具有 16 位数据长度,且表示当方向键分别指定上,下,左,和右方向的方向键时,所要选择的按钮的 ID。

[0184] 模块 `normal_state_info()`,模块 `selected_state_info()`,和模块 `activated_state_info()` 表示分别与正常状态,选定状态,和激活状态中的按钮相关的信息。

[0185] 首先讨论模块 `normal_state_info()`。域 `normal_start_object_id_ref` 和域 `normal_end_object_id_ref`,每一个均具有 16 位数据长度,表示用于在正常状态中,指定按钮动画的第一个对象和最后一个对象的 ID。也就是说,域 `normal_start_object_id_ref` 和域 `normal_end_object_id_ref`,能够指定用于为相应 ODS 显示按钮动画的按钮图像(即,动画画面)。

[0186] 下一个标记 `normal_repeat_flag` 具有 1 位数据长度,且表示是否重复按钮动画。例如,如果标记 `normal_repeat_flag` 为 0,则不重复动画。如果标记 `normal_repeat_flag` 为 1,则重复该动画。

[0187] 下一个标记 `normal_complete_flag` 具有 1 位数据长度,且根据本发明的一个实施例进行定义。当按钮状态从正常状态改变至选定状态时,标记 `normal_complete_flag` 控制动画操作。

[0188] 更具体地说,如果标记 `normal_complete_flag` 为 1,则当按钮状态从正常状态改变至选定状态时,显示正常状态下定义的所有动画图像。更具体地说,如果标记 `normal_complete_flag` 为 1,则当显示正常状态下动画的同时,响应于从正常状态至选定状态的改

变状态的指令,显示动画画面,该动画画面从一接收到该指令所显示的动画画面开始,直到域 `normal_end_object_id_ref` 所指示的动画画面。

[0189] 如果标记 `normal_complete_flag` 为 1,且如果意味着重复动画的标记 `normal_repeat_flag` 为 1,则显示动画画面,该动画画面从一接收到该指令所显示的动画画面开始,直到域 `normal_end_object_id_ref` 所指示的动画画面。

[0190] 在这种情况下,即使该按钮在未选定状态,或在显示动画时该按钮自身关闭,也能够持续显示动画,直到域 `normal_end_object_id_ref` 所指示的动画画面为止,然后,改变该按钮状态。

[0191] 在例如,由根据域 `selection_time_out_pts` 定义的按钮的选择无效,或根据域 `user_time_out_duration` 定义的菜单自动初始化的情况下,按钮处于未选定状态。

[0192] 相反,如果 `normal_complete_flag` 为 0,则当按钮状态从正常状态改变至选定状态时,一收到改变按钮状态而不需显示动画的指令,就停止正常状态下按钮定义的动画的显示,直到域 `normal_end_object_id_ref` 所指示的动画画面为止,然后,显示选定状态的按钮。

[0193] 现在,讨论模块 `selected_state_info()`。模块 `selected_state_info()` 包括,与模块 `normal_state_info()` 中的域对应于的域,以及指定声音的域 `selected_state_sound_id_ref`。域 `selected_state_sound_id_ref` 具有 8 位数据长度,且表示选定状态按钮所要回放的声音文件。例如,当按钮状态从正常状态改变至选定状态时,使用该声音文件作为特效声音。

[0194] 域 `selected_start_object_id_ref` 和域 `selected_end_object_id_ref`,每一个均具有 16 位数据长度,表示用于指定选定状态中按钮动画的第一个对象和最后一个对象的 ID。

[0195] 下一个标记 `selected_repeat_flag` 具有 1 位数据长度,且表示是否重复该按钮动画。例如,如果标记 `selected_repeat_flag` 为 0,则不重复该动画。如果标记 `selected_repeat_flag` 为 1,则重复该动画。

[0196] 与 `normal_complete_flag` 相同,下一个标记 `selected_complete_flag` 具有 1 位数据长度,且与 `normal_complete_flag` 一起,根据本发明的一个实施例进行定义。当按钮状态从选定状态改变至其他状态时,标记 `selected_complete_flag` 控制动画操作。也就是说,在按钮状态从选定状态改变为激活状态或从选定状态改变为正常状态时,使用标记 `selected_complete_flag`。

[0197] 与标记 `normal_complete_flag` 相同,如果标记 `selected_complete_flag` 为 1,则当按钮状态从选定状态改变为其他状态时,显示选定状态中定义的所有动画图像。更具体地说,如果标记 `selected_complete_flag` 为 1,则当显示选定状态下动画的同时,响应于从选定状态至其他状态的改变状态的指令,显示动画画面,该动画画面从一接收到该指令所显示的动画画面开始,直到域 `selected_end_object_id_ref` 所指示的动画画面。

[0198] 如果标记 `selected_complete_flag` 为 1,且如果意味着重复动画的标记 `selected_repeat_flag` 为 1,则显示动画画面,该动画画面从一接收到该指令所显示的动画画面开始,直到域 `selected_end_object_id_ref` 所指示的动画画面。

[0199] 在这种情况下,即使该按钮在未选定状态,或在显示动画时该按钮自身关闭,也能

够持续显示动画,直到域 selected_end_object_id_ref 所指示的动画画面为止,然后,改变该按钮的状态。

[0200] 在例如,由根据域 selection_time_out_pts 定义的按钮的选择无效,或根据域 user_time_out_duration 定义的菜单自动初始化的情况下,按钮处于未选定状态。

[0201] 相反,如果 selected_complete_flag 为 0,则当按钮状态从选定状态改变至其他状态时,一收到改变按钮状态而不需显示动画的指令,就停止选定状态下按钮定义的动画的显示,直到域 selected_end_object_id_ref 所指示的动画画面为止,然后,显示其他状态的按钮。

[0202] 在下一个模块 activated_state_info() 中,与模块 normal_state_info() 和模块 selected_state_info() 相反,在按钮状态从激活状态改变为其他状态时,并未定义表示是否重复动画的标记,或用于控制动画操作的标记。在按钮状态改变为激活状态时,执行分别给激活按钮的功能,且按钮处于激活状态的时间被认为是 非常短的。通常,控制按钮从激活状态改变为其他状态并不是最好的。因此,在模块 activated_state_info() 中,可以省略上述两个标记。然而,也可以对它们进行定义。

[0203] 在模块 activated_state_info() 中,域 activated_state_sound_id_ref 具有 8 位数据长度,且表示激活状态下按钮所要回放的声音文件。域 activated_start_object_id_ref 和域 activated_end_object_id_ref,每一个均具有 16 位数据长度,表示分别用于指定激活状态中该按钮动画的第一个动画画面和最后一个动画画面的 ID。

[0204] 下一个域 number_of_navigation_commands 具有 16 位数据长度,且表示该按钮中嵌入的命令的数目。以与域 number_of_navigation_commands 所表示数目相同的次数,重复下一个 FOR 语句的循环,然后,定义该按钮所执行的命令 navigation_command()。换句话说,从一个按钮能够执行多个命令。

[0205] 现在,给出当前 BD-ROM 标准中定义的 PG 流和 IG 流的解码器模型的说明。附图 29 是对 PG 流或 IG 流适用的解码器模型。

[0206] 首先在播放器中置入光盘,然后,从光盘上读取索引文件“index.bdmv”和电影对象文件“MovieObject.bdmv”,并显示顶层菜单。当根据顶层菜单指定所要回放的标题时,通过电影对象文件中的相应导航命令,调用用于回放指定标题的播放列表文件。然后,根据播放列表文件的说明,从光盘读取播放列表所请求的剪辑 AV 流文件,即,MPEG2 传输流。

[0207] 将传输流作为 TS 包提供给 PID 滤波器 100,并分析该 PID。PID 滤波器 100 确定所提供的 TS 包是否存储视频数据,音频数据,菜单数据,或字幕数据。如果 PID 表示菜单数据,即,IG 数据,或字幕数据,即,PG 数据,则附图 29 所示出的解码器模型的配置变为有效。

[0208] 在 PID 滤波器 100 中,在传输流中选择与解码器模型相一致的,存储数据的 TS 包,并将其存储在传输缓冲器 (TB) 101 中。然后,在 TB101 中,提取 TS 包有效负载中存储的数据。当能够重构为 PES 包的数据的数目存储在 TB101 中时,基于 PID 对 PES 包进行重构。也就是说,将划分为 TS 包的段统一为一个 PES 包。

[0209] 从根据段重构的 PES 包中去除 PES 的头部,且将该 PES 包作为基本码流提供给解码器 102,且临时存储在编码数据缓冲器 (CDB) 110 中。在 CDB110 所存储的基本码流中,根据 STC,如果存在已到达基于 STC 的 DTS 所指的时间的基本码流,则从 CDB110 中读取该基本码流,并将其传输至流图形处理器 111。然后将该码流解码为多个段。

[0210] 流图形处理器 111 在解码对象缓冲器 (DB) 112 或合成缓冲器 (CB) 113。如果段的类型是 DTS 中提供的类型, 诸如 PCS, ICS, WDS, 或 ODS, 则根据 DTS 所表示的时间, 将该段存储在 DB112 或 CB113 中。如果段的类型不是 DTS 中提供的类型, 诸如 PDS, 则立即将该段存储在 CB113 中。

[0211] 图形控制器 114 控制多个段。例如, 如果解码器模型与 PG 数据相一致, 则图形控制器 114 根据 PTS 所表示的时间, 从 CB113 中读取 PCS, 并且还读取 PCS 所引用的 WDS 和 PDS。图形控制器 114 还从 DB112 中读取 PCS 所涉及的 ODS。然后, 图形控制器 114 对所读取的 PCS, WDS 和 ODS 进行解码, 从而形成用于显示字幕的数据, 然后将该数据写入图形面板 103。

[0212] 同样, 如果解码器模型是与 IG 数据相一致的模型, 则图形控制器 114 根据对应于 ICS 的 PTS 所表示的时间, 从 CB113 中读取 ICS, 并且还读取该 ICS 所涉及的 PDS。图形控制器 114 还从 DB112 中读取 ICS 所涉及的 ODS。然后, 图形控制器 114 对所读取的 ICS 和 ODS 进行解码, 从而形成用于显示菜单屏幕的数据, 诸如按钮图像, 并将该数据写入图形面板 103。

[0213] 图形控制器 114 还对从 CB113 中读取的 PDS 进行解码, 从而形成附图 13 中所讨论的颜色调色板表格, 并将该颜色调色板表格写入颜色查找表 (CLUT) 104。

[0214] 在预定时间, 诸如画面时间, 读取写入图形面板 103 的图像, 并将颜色信息添加至 CLUT104 的颜色调色板表格所涉及的图像, 以便形成输出图像数据。然后, 输出该输出图像数据。根据播放器的说明书, 可以进一步将基于 CLUT104 颜色调色板表格的颜色信息, 添加至从图形面板 103 读取的图像数据, 以便形成字幕图像数据, 然后将所得到的字幕图像数据写入至画面存储器。

[0215] 在上述解码器模型中, 对于 PG 数据和 IG 数据, 将多个段存储在 CDB110 中。然后, 在段中定义的 PTS 所表示的时间, 对每一段进行解码, 并将其存储在 DB112 或 CB113 中, 然后, 在段中定义的 PTS 所表示的时间将其输出。这样, 当基于在 BD-ROM 标准中定义的解码器模型涉及播放器时, 将会考虑到与 CDB110 中所存储的段相对应的存储 DTS 和 PTS 的存储器的准备。附图 29 中的虚线, 如 PTS/DTS 缓冲器 150, 指出该存储器。在当前 BD-ROM 标准中, 并未定义 PTS/DTS 缓冲器 150。

[0216] 基于解码器模型控制播放器的控制器, 根据 PTS/DTS 缓冲器 150 中存储的 DTS, 指定用于 CDB110 中所存储段的解码时间, 并根据 PTS/DTS 缓冲器 150 中存储的 PTS, 指定 DB112 或 CB113 中解码并存储一个段的输出时间。

[0217] 附图 30 描述了 PTS/DTS 缓冲器 150 中所存储数据的结构的实例。在 PTS/DTS 缓冲器 150 中, 对应于 CDB110 所存储的段, 存储 DTS 和 PTS, 且存储用于识别该段的段 ID。根据规则, DTS 和 PTS 中的每一个均具有 33 位数据长度。

[0218] 在 PTS/DTS 缓冲器 150 中, 也可以为每一段存储其他类型的数据。在附图 30 所示出的实例中, 存储与段 ID 所指示的段的版本相关的版本信息和与段类型相关的信息。与段类型相关的信息表示就 PG 数据来说, 该段是否为 PCS, PDS, WDS, 或 ODS, 且就 IG 数据来说, 该段是否为 ICS, PDS, 或 ODS。

[0219] 在附图 30 所示出的域中, 如果, 例如, 段 ID 的数据长度为 16 位, 如果版本信息的数据长度为 8 位, 且如果段类型的数据长度为 4 位, 则每段的数据尺寸为 94 位, 即, 按照字

节如果四舍五入为 12 字节。

[0220] 在上述解码器模型中,根据时间点的更新,清除不同的缓冲器和面板。当从 CDB110 中读取,且在流图形处理器 111 中进行解码的 PCS 或 ICS 的模块 `composition_descriptor()` 中的域 `composition_state`,表示对应显示集是时间点起始显示集时,由图形控制器 114 清除 DB112, CB113, 图形面板 103, 和 CLUT104。根据时间点的更新,也可以清除 PTS/DTS 缓冲器 150。

[0221] 下面将说明时间点的定义。如上所述,在 PG 流或 IG 流中,根据时间点起始显示集定义时间点。

[0222] 对于 PG 流,将下面项目 (1) 和 (2) 中显示集的范围较短的一个,定义为一个时间点:

[0223] 项目 (1): 一组显示集从作为 PCS 中定义的时间点起始显示集的显示集开始,直到作为 PCS 中定义的下一个时间点起始显示集的显示集之前的一个显示集为止;以及

[0224] 项目 (2): 一组显示集从作为 PCS 中定义的时间点起始显示集的显示集开始,直到与相应 PG 数据相关的播放列表结束为止。

[0225] 更具体地说,就项目 (1) 来说,以一组显示集定义一个时间点,这一组显示集从定义时间点起始显示集的 PCS 的 PTS 开始,直到定义下一个时间点起始显示集的 PCS 的 PTS 为止。同样,就项目 (2) 来说,以一组显示集定义一个时间点,这一组显示集从定义时间点起始显示集的 PCS 的 PTS 开始,直到与该组显示集相关的播放列表结束为止。

[0226] 对于 IG 流,将下面项目 (3)、(4) 和 (5) 中显示集的范围较短的一个,定义为一个时间点:

[0227] 项目 (3): 一组显示集从作为 ICS 中定义的时间点起始显示集的显示集开始,直到 ICS 的域 `composition_time_out_pts` 中定义的时间为止;

[0228] 项目 (4): 一组显示集从作为 ICS 中定义的时间点起始显示集的显示集开始,直到作为 ICS 中定义的下一个时间点起始显示集的显示集之前的一个显示集为止;以及

[0229] 项目 (5): 一组显示集从作为 ICS 中定义的时间点起始显示集的显示集开始,直到与该 IG 数据相关的播放列表结束为止。

[0230] 附图 31 是描述图形解码器 102 在 PG 数据显示集的单元中所执行的处理的流程图。在步骤 S10 中, PID 滤波器 100 基于 PID 从 TS 包中获得存储 PG 数据的 TS 包,并将所获得的 TS 包存储在 TB101 中。然后,在步骤 S11 中,当存储在 TB101 中的能够重构 PES 包的 TS 包的数目已经足够的时候,基于 PID 从 TS 包的有效负载中提取数据,以便重构 PES 包。然后,从 PES 包中去除 PES 的头部,并将 PES 包作为基本码流传输给 CDB110。

[0231] 在步骤 S12 中,流图形处理器 111 将从 PES 包中提取的基本码流扩展为段。然后在步骤 S13 到 S17 中,分别确定所扩展的段是否为 PCS, WDS, PDS, ODS, 或结束段。然后,根据所确定的结果执行处理。如附图 19 或附图 21 所讨论的,基于语法中第一个模块 `segment_description()` 的第一个域 `segment_type`, 确定段的类型。虽然并未示出,但以类似的方式,也可以确定 WDS 和 ODS 的段的类型。

[0232] 在步骤 S13 中,确定该段是否是 PCS。如果发现该段是 PCS,则根据附图 19 所讨论的 PCS 的语法中模块 `composition_descriptor()` 中的域 `composition_state`, 处理进行至步骤 S19, 从而确定包括该 PCS 的显示集是否为时间点起始显示集。

[0233] 如果在步骤 S19 中发现该显示集是时间点起始显示集,则处理进行至步骤 S20。在步骤 S20 中,清除涉及字幕显示的缓冲器,例如 DB112 和 CB113。这样,也可以清除 CDB110。在步骤 S21 中,根据 DTS 所表示的时间,将 PCS 写入在步骤 S20 清除的 CB113 中。然后,处理返回至步骤 S10,且获得下一个包。

[0234] 如果在步骤 S19 中确定包括该 PCS 的显示集不是时间点起始显示集,则处理进行至根据 DTS 所表示的时间,将 PCS 写入 CB113 的步骤 S22。这样,将 PCS 写入 CB113 的预定空间。

[0235] 如果在步骤 S13 中确定该段不是 PCS,则处理进行至步骤 S14,从而确定该段是否为 WDS。如果发现该段是 WDS,则处理进行至步骤 S23。在步骤 S23 中,根据 DTS 所表示的时间,将 WDS 写入 CB113 中。然后,处理返回至步骤 S10,并获得下一个包。如果在步骤 S14 中确定该段不是 WDS,则处理进行至步骤 S15,从而确定该段是否为 PDS。

[0236] 如果发现该段是 PDS,则处理进行至步骤 S24。在步骤 S24 中,立即将 PDS 写入 CB113 中。由于 PDS 并不具有 DTS,但具有 PTS,所以,在将其确定为 PDS 之后,立即将其写入 CB113。然后,处理返回至步骤 S10,并获得下一个包。如果在步骤 S15 中确定该段不是 PDS,则处理进行至步骤 S16,从而确定该段是否为 ODS。

[0237] 如果发现该段是 ODS,则处理进行至步骤 S25。在步骤 S25 中,根据 DTS 所表示的时间,解码 ODS 中存储的游程长度处理的位图数据,并将其写入 DB112。然后,处理返回至步骤 S10,并获得下一个包。如果在步骤 S16 中确定该段不是 ODS,则处理进行至步骤 S17,从而确定该段是否为作为显示集结束的结束段。

[0238] 如果发现该段是结束段,则结束关于该显示集的处理序列。

[0239] 如果在步骤 S17 中确定该段不是结束段,则处理进行至步骤 S18。也就是说,确定发生某一误差,并执行预定误差处理。

[0240] 在上述配置中,根据 PCS 中定义的 PTS,从 CB113 输出用于显示字幕的 PCS,且显示该字幕之后,从 CB113 中删除 PCS。然而,根据该方案,如果用户给出关闭字幕显示的指令,则在时间到达 PCS 中定义的 PTS 时,很难重现该字幕,即使用户试图通过开启字幕的显示来重现该字幕。

[0241] 因此,从 DB112 中单独提供用于存储当前激活 ODS 的缓冲器,并且,从 CB113 中单独提供用于存储当前激活 PCS 和 PDS 的缓冲器。响应于开启字幕显示的指令,利用 DB112 和 CB113 中单独提供的缓冲器所存储的 ODS,PCS,和 PDS,形成字幕图像和颜色调色板表格,并将其分别写入图形面板 103 和 CLUT104。

[0242] “激活”意味着 PTS 所指示的相应段的显示的状态,且更具体地说,从某一 PCS 中定义的 PTS 开始,到下一个用于显示字幕的 PCS 中定义 PTS 的一段时间。

[0243] 也就是说,图形控制器 114 根据 PCS 中定义的 PTS,将 CB113 输出的 PCS 提供给图形面板 103 和 CLUT104,并且还保留 CB113 中单独提供的缓冲器中的 PCS,直到到达下一个 PCS 中定义的 PTS 的时间。上述内容同样适用于 ODS。图形控制器 114 根据 PTS,将 DB112 输出的 ODS 提供给图形面板 103,并且还保留 DB112 中单独提供的缓冲器中的 ODS,直到到达下一个 PCS 中定义 PTS 的时间。

[0244] 根据该配置,如果在根据用户操作关闭字幕显示之后重构字幕的显示,则与开启字幕显示的指令同步,立即重新开启字幕的显示。

[0245] 也就是说,响应于关闭字幕显示之后开启字幕显示的指令,从 CB113 单独提供的缓冲器中读取 PCS,且还从 DB112 单独提供的缓冲器中读取 ODS,并且利用所读取的 PCS 和 ODS,形成字幕图像。然后,将字幕图像写入图形面板 103。上述内容同样适用于 PDS。响应于开启字幕显示的指令,从 CB113 单独提供的缓冲器中读取 PDS,且利用所读取的 PDS,形成颜色调色板表格。然后,将颜色调色板表格写入 CLUT104。

[0246] 附图 32 是描述图形解码器 102 在 IG 数据显示集的单元中所执行的处理的流程图。附图 32 所示出的处理与附图 31 中示出的对 PG 数据的处理相类似,除了对 PG 数据所特有的 WDS 的处理之外。

[0247] 在步骤 S30 中,PID 滤波器 100 基于 PID 从 TS 包中获得存储 IG 数据的 TS 包,并将所获得的 TS 包存储在 TB101 中。然后,在步骤 S31 中,从 TB101 中存储的 TS 包的有效负载中提取数据,并重构 PES 包。然后,从 PES 包中去除 PES 的头部,并将 PES 包作为基本码流传输给 CDB110。

[0248] 在步骤 S32 中,流图形处理器 111 将从 PES 包中提取的基本码流扩展为段。然后在步骤 S33 到 S36 中,分别确定所扩展的段是否为 ICS, PDS, ODS,或结束段,并根据所确定的结果执行处理。如附图 24 所讨论的,基于语法中第一个模块 `segment_description()`,确定段的类型。虽然并未示出,但以类似的方式,也可以确定 ODS 的段的类型。

[0249] 在步骤 S33 中,确定该段是否是 ICS。如果发现该段是 ICS,则根据附图 24 所讨论的 ICS 的语法中模块 `composition_descriptor()`,处理进行至步骤 S38,从而确定包括该的显示集是否为时间点起始显示集。

[0250] 如果在步骤 S38 中发现该显示集是时间点起始显示集,则处理进行至步骤 S39。在步骤 S39 中,清除涉及菜单显示的缓冲器,例如 DB112 和 CB113。这样,也可以清除 CDB110。然后,在步骤 S40 中,根据 DTS 所表示的时间,将 ICS 写入在步骤 S39 清除的 CB113 中。然后,处理返回至步骤 S30,且获得下一个包。

[0251] 如果在步骤 S38 中确定包括该 ICS 的显示集不是时间点起始显示集,则处理进行至步骤 S41。在步骤 S41 中,根据 DTS 所表示的时间,将 ICS 写入 CB113。这样,将 ICS 写入 CB113 的预定空间。

[0252] 如果在步骤 S33 中确定该段不是 ICS,则处理进行至步骤 S34,从而确定该段是否为 PDS。如果该段不是 PDS,则处理进行至步骤 S35,从而确定该段是否为 ODS。如果该段不是 ODS,则处理进行至步骤 S36,从而确定该段是否为结束段。如果在步骤 S34 中发现该段是 PDS,则在步骤 S42 中立即将 PDS 写入 CB113。如果在步骤 S35 中发现该段是 ODS,则在步骤 S43 中根据 DTS 所表示的时间,解码 ODS 并将其写入 DB112。

[0253] 以下,将讨论适用上述解码器模型的回放设备。附图 33 描述了基于上述解码器模型的回放设备 1 的配置。回放设备 1 包括存储驱动器 50, PID 滤波器 51, AV 解码器 52,和控制器 53。存储驱动器 50 能够在其中安装上述 BD-ROM,并对其进行播放。

[0254] 控制器 53 包括中央处理单元 (CPU),预存储 CPU 上所运行程序的 ROM,当 CPU 执行程序时用作工作存储器的随机存取存储器 (RAM) 等。根据该配置,控制器 53 控制回放设备 1 的所有操作。

[0255] 提供具有用户界面的回放设备 1,该用户界面可以向用户提供预定控制信息,并且还响应于用户操作输出控制信号,虽然上述用户界面并未示出。例如,使用通过预定无

线通信单元,例如,红外通信单元,执行与回放设备 1 的远程通信的远程控制器作为用户界面。在该远程控制器上,提供多个输入单元,该多个输入单元诸如例如,横向键之类的指定上,下,左,和右方向的方向键,数字键盘,分配有多个功能的功能键盘。

[0256] 远程控制器根据在一个输入单元上执行的操作,产生一个控制信号,并将所产生的信号调制为,例如,红外信号,然后,传输该红外信号。回放设备 1 通过红外接收器(未示出)接收该红外信号,并将该红外信号解调为电信号,从而重构源控制信号。然后将该控制信号提供给控制器 53。响应于该控制信号,控制器 53 根据程序控制回放设备 1 的操作。

[0257] 用户界面并不局限于远程控制器,但可以在回放设备 1 的操作面板上提供一个开关组。作为选择,可以为回放设备 1 提供通过,例如局域网,执行通信的通信单元作为用户界面,且可以将通过该通信单元从外部计算机提供的信号,作为用户界面所提供的控制信号提供给控制器 53。

[0258] 在回放设备 1 的非易失性存储器中存储关于语言设置的初始信息。从非易失性存储器中读取语言设置的初始信息,并当例如,回放设备 1 通电时,将其提供给控制器 53。

[0259] 当将光盘载入存储驱动器 50 时,控制器 53 通过存储驱动器 50 从光盘上读出文件“index.bdmv”和文件“MovieObject.bdmv”,并基于所读取文件的说明,读出目录“PLAYLIST”中的播放列表文件。然后控制器 53 通过存储驱动器 50 从光盘上读出播放列表文件中包括对播放项所涉及的剪辑 AV 流文件。如果播放列表包括子播放项,则控制器 53 还通过存储驱动器 50 从光盘上读出于播放项所涉及的剪辑 AV 流文件和字幕数据。

[0260] 下文中,对应于子播放项的剪辑 AV 流被称为“子剪辑 AV 流”,且对应于与子播放项相关的主播放项的剪辑 AV 流被称为“主剪辑 AV 流”。

[0261] 然后,由解调器(未示出)解调从存储驱动器 50 输出的数据,且由误差校正器(未示出)对其执行误差校正处理,以便重构多路复用流。多路复用流是一传输流,该传输流中在通过 PID 识别数据类型和数据顺序之后,将数据划分为具有预定尺寸的较小的部分,并将较小的部分进行时分多路复用。PID 滤波器 51 根据基于 PID 类型对数据进行分类。更具体地说,PID 滤波器 51 将主剪辑 AV 流包提供给缓冲器 60,将子剪辑 AV 流包提供给缓冲器 61,将声音数据包提供给缓冲器 62,以及将文本数据包提供给缓冲器 63。

[0262] 在控制器 53 的控制下,以逐包处理为基础,从缓冲器 60 中读取存储在缓冲器 60 中的主剪辑 AV 流包,并将其提供给 PID 滤波器 64。PID 滤波器 64 基于包的 PID 将包分类至视频流包,PG 流包,IG 流包,和音频流包。

[0263] 在控制器 53 的控制下,以逐包处理为基础,从缓冲器 61 中读取存储在缓冲器 61 中的子剪辑 AV 流包,并将其提供给 PID 滤波器 90。PID 滤波器 90 基于包的 PID 将包分类至视频流包,PG 流包,IG 流包,和音频流包。

[0264] 将 PID 滤波器 64 所分类的视频流包和 PID 滤波器 90 所分类的视频流包,提供给 PID 滤波器 65,并根据 PID 对它们进行分配。也就是说,PID 滤波器 65 将 PID 滤波器 64 输出的主剪辑 AV 流包提供给第一视频解码器 69,并将 PID 滤波器 90 输出的子剪辑 AV 流包提供给第二视频解码器 72。

[0265] 第一视频解码器 69 从所提供包的有效负载中提取预定视频流,并对所提取视频流的 MPEG2 压缩码进行解码。然后,第一视频解码器 69 将已解码的数据输出至第一视频面板产生器 70,并产生视频面板。这样,通过例如,将基带数字视频数据写入画面存储器,产生

视频面板。将第一视频面板产生器 70 中产生的视频面板,提供给视频数据处理器 71。

[0266] 分别由第二视频解码器 72 和第二视频面板产生器 73,执行类似于第一视频解码器 69 和第一视频面板产生器 70 所执行的处理,以便解码视频流,并产生视频面板。将第二视频面板产生器 73 中产生的视频面板,提供给视频数据处理器 71。

[0267] 视频数据处理器 71 将第一视频面板产生器 70 中产生的视频面板和第二视频面板产生器 73 中产生的视频面板,插入至一个画面中,从而产生一个视频面板。通过选择性的使用第一视频面板产生器 70 中产生的视频面板和第二视频面板产生器 73 中产生的视频面板,可以产生一个视频面板。所产生的视频面板,对应于例如,附图 9 中示出的活动图像面板 10。

[0268] 将 PID 滤波器 64 所分类的 PG 流包和 PID 滤波器 90 所分类的 PG 流包,提供给 PID 滤波器 66,且选择 PG 流包的一种类型,并将其提供给 PG 解码器 74。

[0269] PG 解码器 74,对应于附图 29 所讨论的解码器模型中的图形解码器 102。这样,附图 32 所示出的 PID 滤波器 100 和 TB101,分别对应于 PID 滤波器 64 和缓冲器(未示出)。

[0270] PG 解码器 74 从所提供包的有效负载中提取预定 PG 流包,并对 PG 流包进行解码,从而产生显示字幕所用的图形数据。然后,PG 解码器 74 将该图形数据提供给 PID 滤波器 75。也就是说,PG 解码器 74 从所提供包的有效负载中提取数据,并基于 PID 重构 PES 包。如果 DTS 和 / 或 PTS 包括在所重构的 PES 包的头部中,则 PG 解码器 74 提取 DTS 和 / 或 PTS,并将其存储在 PTS/DTS 缓冲器 150 中,且还对有效负载的数据进行解码,从而形成 PCS, WDS, PDS, 和 ODS。在相应 DTS 所表示的时间,将 PCS, WDS, PDS, 和 ODS 均存储在 DB112 或 CB113 中,且在相应 PTS 所表示的时间,将其从 PG 解码器 74 中输出。将 PCS, WDS, PDS, 和 ODS 均提供给 PID 滤波器 75。

[0271] PID 滤波器 75 以如下所述的文本数据的形式,选择预定图形数据和字幕数据,并将所选择的数据提供给 PG 面板产生器 76。PG 面板产生器 76 基于所提供的数据产生 PG 面板,并将 PG 面板提供给视频数据处理器 71。PG 面板对应于例如,附图 9 所讨论的字幕面板 11,且还对应于附图 29 所讨论的图形面板 103 中存储的字幕图像。

[0272] 将 PID 滤波器 64 所分类的 IG 流包和 PID 滤波器 90 所分类的 IG 流包,提供给 PID 滤波器 67,且选择 IG 流包的一种类型,并将其提供给 IG 解码器 77。

[0273] IG 解码器 77,对应于附图 29 所讨论的解码器模型中的图形解码器 102。这样,附图 29 所示出的 PID 滤波器 100 和 TB101,分别对应于 PID 滤波器 67 和缓冲器(未示出)。

[0274] IG 解码器 77 从所提供的 IG 包中提取 ICS, PDS, 和 ODS,并对 ICS, PDS, 和 ODS 进行解码。例如,IG 解码器 77 从所提供包的有效负载中提取数据,并基于 PID 重构 PES 包。如果 DTS 和 / 或 PTS 包括在所重构的 PES 包的头部中,则 IG 解码器 77 提取 DTS 和 / 或 PTS,并将其存储在 PTS/DTS 缓冲器 150 中,且还对有效负载的数据进行解码,从而形成 IG 流包的 ICS, PDS, 和 ODS。在 PTS/DTS 缓冲器 150 所存储的相应 DTS 所表示的时间,将 ICS 和 ODS 进行解码,然后将其存储在 DB112 中。PDS 立即被解码并存储在 CB113 中。

[0275] IG 解码器 77 中,在 PTS/DTS 缓冲器 150 所存储的 PTS 所表示的时间,从 DB112 或 CB113 中读出相应段,并通过预载入缓冲器 78 将其提供给 IG 面板产生器 79。IG 面板产生器 79 基于所提供的段,产生 IG 面板。该 IG 面板对应于,例如,附图 9 所讨论的 IG 面板 12。

[0276] 视频数据处理器 71 包括,例如,附图 11 所讨论的图形处理器,且对视频面板(附图 11 中示出的活动图像面板 10),PG 面板(附图 11 中示出的字幕面板 11),和 IG 面板(附图 11 中示出的 IG 面板 12)进行合并,从而形成一段图像数据,并将该图像数据作为视频信号输出。

[0277] 以下,将讨论在附图 11 中 PG 面板(即,字幕图像)的情况下的细节。调色板 22A,RGB/YCbCr 转换电路 22B,和乘法器 23,对应于附图 29 中所讨论的 CLUT104。将颜色信息添加至调色板 22A 和 RGB/YCbCr 转换电路 22B 中的 PG 面板,且将不透明信息添加至乘法器 23 中的 PG 面板。然后,分别将所得到的 PG 面板与加法器 24 和加法器 28 中的视频面板和 IG 面板,以便输出一段图像数据。

[0278] 将 PID 滤波器 64 所分类的音频流和 PID 滤波器 90 所分类的音频流,提供给 PID 滤波器 68。PID 滤波器 68 切换 PID 滤波器 68 的开关,以便将音频流的一种类型,提供给第一音频解码器 80,并且,将其他类型的音频流提供给第二音频解码器 81。第一音频解码器 80 和第二音频解码器 81 中解码的音频流在加法器 82 中进行合并,且进一步还在加法器 83 中将所加得的音频流与从缓冲器 62 中读取的声音数据相合并。然后,输出所得到的音频信号。

[0279] 在文本-ST 合成单元 91 中处理从缓冲器 63 中读取的文本-ST,并将其提供给 PID 滤波器 75。

[0280] 现在,下面将说明本发明的一个实施例。在 BD-ROM 标准中,定义 PG 数据或 IG 数据中所使用的段的数目,和上述解码器模型中所使用的缓冲器的缓冲区尺寸。在本发明的一个实施例中,提供数据结构和记录该数据结构的记录介质,该数据结构满足所定义的条件,例如,PG 数据或 IG 数据所使用的段的数目和缓冲器的缓冲区尺寸。在本发明的一个实施例中,提供了用于产生该数据结构的创作设备和方法,用于在记录介质上记录该数据结构的记录设备和方法,以及用于制造该记录介质的制造设备和方法。在本发明的一个实施例中,提供了用于确定所产生数据是否满足 BD-ROM 标准所定义的条件,诸如 PG 数据或 IG 数据所使用的段的数目和缓冲器的缓冲区尺寸,的验证设备和方法。

[0281] 下面,将简要讨论本发明相关的 BD-ROM 标准中 PG 数据或 IG 数据所定义的条件。对 PG 数据和 IG 数据中的每一个,定义在一个时间点内段的数目的上限。由于该上限,就 PG 数据来说,能够限制一个时间点内所使用的字幕图像数据的数目,并且,就 IG 数据来说,能够定义一个时间点内所使用的页,命令或按钮图像数据的数目。

[0282] 一个时间点内 PG 数据的段的数目的上限为,例如,下面在附图 34A 所示的:8 个 PCS,1 个 WDS,8 个 PDS,64 个 ODS,以及 8 个结束段。一个时间点内 IG 数据的段的数目的上限为,例如,下面在附图 34B 所示的:1 个 ICS,256 个 PDS,4096 个 ODS,以及 1 个结束段。

[0283] 基于例如,段的数目的上限,如上所述,确定 CDB110,DB112,CD113 和 PTS/DTS 缓冲器 150 的最小容量。

[0284] 例如,对 CDB110,定义 CDB110 中所能够存储的段的数目的上限。也就是说,如果段的数目超过图形流处理器 111 的处理能力,且下一个阶段存储在 CDB110 中,则可以打乱处理。例如,如果超过 DB112 和 CB113 中所能存储段数目的段的数目,存储在 CDB110 中,则根据段的 DTS 和 PTS 之间的关系,不能将从 CDB113 中读出并在 DTS 所表示的时间对其进行解码的段,存储在 DB112 和 CB113 中。

[0285] 因此,基于 DB112 和 CB113 中存储的段的数目,确定 CDB110 中存储的段的数目的上限。如同一个特定的实例,对于 PG 数据,基于一个时间点内 PG 数据的段的数目的上限,如附图 34A 所讨论的,确定 $8(\text{PCS})+1(\text{WDS})+8(\text{PDS})+64(\text{ODS})+8(\text{结束段})=89$ 为 CDB110 中所能存储的段的数目的上限。实际上,对该上限添加某个余量,且将 CDB110 中能够存储的 PG 数据的段的数目的上限,设置为,例如,128。

[0286] 对 IG 数据应用相同的处理。也就是说,对于 IG 数据,基于一个时间点内 IG 数据的段的数目的上限,如附图 34B 所讨论的,确定 $1(\text{ICS})+256(\text{PDS})+4096(\text{ODS})+1(\text{结束段})=4354$ 为 CDB110 中所能存储的段的数目的上限。实际上,对该上限添加某个余量,且将 CDB110 中能够存储的 IG 数据的段的数目的上限,设置为,例如,5120。

[0287] 基于 CDB110 中能够存储的段的数目的上限,确定用于 PG 数据的 PTS/DTS 缓冲器 150 的最小容量。基于 DB112 和 CB113 中能够存储的段的数目,确定 CDB110 中能够存储的段的数目的上限,如上所述。因此,下面将确定用于 PG 数据 PTS/DTS 缓冲器 150 的最小容量。

[0288] 在形成 PG 数据的各段中,由于 END 段不具有 DTS,并且不存储在 DB112 或 CB113 中,所以为了计算 PTS/DTS 缓冲器 150 的最小容量不需对 END 段进行计数。由于 PDS 不具有 DTS,但可以存储在 CB113 中,所以可以为了计算 PTS/DTS 缓冲器 150 的最小容量而对 PDS 进行计数。因此,基于 PCS, WDS, PDS, 和 ODS 的数目的上限,评估与 PG 数据相一致的解码器模型中 PTS/DTS 缓冲器 150 的最小容量。

[0289] 如果将附图 30 所示出数据结构的所有的域存储在每一个段中,则数据尺寸变为 $(8+1+8+64) \times 12 \text{ 字节} = 972 \text{ 字节}$,且与 PG 数据相一致的解码器模型中 PTS/DTS 缓冲器 150 的最小容量被评估为 972 字节。实际上,对该最小容量添加某个余量,且将 PTS/DTS 缓冲器 150 中的最小容量评估为约 1.6KB。

[0290] 与 PG 数据的情况相同,基于 CDB110 中能够存储的段的数目的上限,确定对于 IG 数据的 PTS/DTS 缓冲器 150 的最小容量。如上所述,基于 DB112 和 CB113 中能够存储的段的数目,确定 CDB110 中能够存储的段的数目的上限。因此,下面将确定对于 IG 数据的 PTS/DTS 缓冲器 150 的最小容量。

[0291] 在形成 IG 数据的各段中,由于结束段不具有 DTS,且不存储在 DB112 或 CB113 中,所以可以为了计算 PTS/DTS 缓冲器 150 的最小容量而不对结束段进行计数。由于 PDS 不具有 DTS,但可以存储在 CB113 中,所以可以为了计算 PTS/DTS 缓冲器 150 的最小容量而对 PDS 进行计数。因此,基于 ICS, PDS, 和 ODS 的数目的上限,评估与 IG 数据相一致的解码器模型中 PTS/DTS 缓冲器 150 的最小容量。

[0292] 如果将附图 30 所示出数据结构的所有的域存储在每一个段中,则数据尺寸变为 $(1+256+4096) \times 12 \text{ 字节} = 52236 \text{ 字节}$,且与 IG 数据相一致的解码器模型中 PTS/DTS 缓冲器 150 的最小容量被评估为 52236 字节。实际上,对该最小容量添加某个余量,且将 PTS/DTS 缓冲器 150 的最小容量评估为约 64KB。

[0293] 以下将根据本发明的一个实施例,描述 PG 数据或 IG 数据的数据结构。该实施例中,提供 PG 数据和 / 或 IG 数据的数据结构,该数据结构满足以下条件,即,在某一时间点,从光盘中读取并存储在作为解码缓冲器的 CDB110 中的段的数目,不超过 CDB110 中能够存储的段的数目的上限。

[0294] 以下将参考附图 35, 根据本发明的一个实施例, 讨论 PG 数据或 IG 数据的数据结构的概念。现在假定 CDB110 中能够存储的段的数目的上限为 N 。附图 35 示意性的描述了在某一时间点 CDB110 的状态。在 CDB110 中, 如附图 35 所示, 已存储作为能够存储在 CDB110 中的段的数目的上限的 N 段, 诸如段 $\text{Seg}_{\#n}$, $\text{Seg}_{\#n+1}$, $\text{Seg}_{\#n+2}$, $\dots$, $\text{Seg}_{\#n+N-2}$, $\text{Seg}_{\#n+N-1}$ 。 $\text{Seg}_{\#n}$ 和 $\text{Seg}_{\#n+1}$ 是将提供给 CDB110 的段。

[0295] 附图 35 中, 段的排列的顺序与提供段的顺序相对应, 且向右边的时间较早。例如, 段 $\text{Seg}_{\#n}$ 是存储在 CDB110 中的段中, 在最早的时间存储在 CDB110 中的段。段 $\text{Seg}_{\#n+1}$ 和 $\text{Seg}_{\#n}$ 是还未存储在 CDB110 中, 且以后将要存储在 CDB110 中的段。段 $\text{Seg}_{\#n}$ 是下一个将要存储在 CDB110 中的段。

[0296] 对每一个段指定 DTS。在附图 35 示出的状态中对最陈旧的段 $\text{Seg}_{\#n}$ 指定 $\text{DTS}_{\#n}$ 。将要提供给 CDB110 的段的顺序, 对应于为各段指定的 DTS 的时序顺序。

[0297] 如果在该状态下将 $\text{Seg}_{\#n}$ 提供给 CDB110, 则为了满足以下条件, 即, CDB110 中存储的段的数目, 不超过 CDB110 中存储的段的数目的上限 N , 在将 $\text{Seg}_{\#n}$ 提供给 CDB110 之前, 从 CDB110 中输出至少一个段。

[0298] 更具体地说, 如果将段 $\text{Seg}_{\#n}$ 存储在 CDB110 的时间 $\text{RT}_{\#n}$, 晚于在最早的时间从 CDB110 中输出的段 $\text{Seg}_{\#n}$ 所指定的 $\text{DTS}_{\#n}$, 则当将段 $\text{Seg}_{\#n}$ 提供给 CDB110 时, CDB110 中存储的段的数目并未超过段的数目的上限 N 。

[0299] 换句话说, 提供以下数据结构。现在假定基于某一段 x , 在段 x 之前 CDB110 中存储的段的最大值限制在 N 内。这样, 表示为 N 段指定的 DTS 中最早时间的 DTS, 应早于 CDB110 中存储段 x 的时间。根据该数据结构, 可以一直满足由 CDB110 中的段的数目的上限所定义的条件。

[0300] 现在, 考虑 CDB110 中存储段 x 的时间。如附图 33 中示出的, 从由光盘中读取并由存储驱动器 50 输出的一个数据流开始, 在 PID 滤波器 51 中选择一个主剪辑 AV 流, 并将其临时存储至缓冲器 60。然后, 以 TS 逐包处理为基础, 从缓冲器 60 读出主剪辑 AV 流。也就是说, 从缓冲器 60 输出对视频流包, 音频流包, PG 流包, 和 IG 流包进行多路复用的剪辑 AV 流, 作为排列在逻辑结构中的包。

[0301] 在 PID 滤波器 64 中分类从缓冲器 60 中读出的包, 并在 PID 滤波器 65 至 68 中基于 PID 对其进行选择, 且将所选定的包提供给相应的模块。例如, PID 滤波器 64 和 66 对 PG 流进行选择, 且立即将它们形成段, 并将其提供给 PG 解码器 74。然后, 将提供给解码器 74 的段存储在 CDB110 中。

[0302] 附图 36A 和 36B 示意性的描述了将数据流从缓冲器 60 提供给 PG 解码器 74 或 IG 解码器 77 的流程。附图 36A 描述了基于附图 29 所讨论的解码器模型的数据流的流程。在 BD-ROM 标准所定义的解码器模型中, 以每秒 48 兆位 (Mbps) 的传输率, 将数据流提供给 PID 滤波器 100。以 48Mbps 的传输率从 TB101 中输出基于 PID 在 PID 滤波器 100 中进行分类的 TS 包, 并将其提供给 CDB110。

[0303] 附图 36B 描述了将基于附图 36A 所示解码器模型的数据流应用于附图 33 所示回放设备 1 的实例。这样, 对于 PG 数据, 解码器模型的 PID 滤波器 100 和 TB101, 分别对应于 PID 滤波器 64 和 PID 滤波器 66, 且对于 IG 数据, PID 滤波器 100 和 TB101, 分别对应于 PID 滤波器 64 和 PID 滤波器 67。因此, 在 PG 数据的情况下, 以 48Mbps 的传输率从缓冲器 60 输

出的流中选择 PG 包,且在 PID 滤波器 64 和 66 中立即将它们形成段,并以 48Mbps 的传输率将其提供给 PG 解码器 74。将提供给 PG 解码器 74 的段存储在 PG 解码器 74 的解码缓冲器(对应于 CDB110)内。

[0304] 下面将根据附图 37 更具体地进行说明。在附图 37 中,多个方块表示 TS 包,V,A,P,和 I 分别表示视频包,音频包,PG 包,和 IG 包。在附图 37 中,向右边的时间较早。附图 37 的 A 部分表示缓冲器 60 输出的数据流的实例。缓冲器 60 的输出是对视频包,音频包,PG 包,和 IG 包进行多路复用的多路复用包流。

[0305] 如上所述,根据回放设备 1 中 PID 滤波器 64 到 68 中的数据类型,对包括缓冲器 60 输出的不同类型包的包流进行选择,并且还将 PG 数据和 IG 数据重构为 PES 包(未示出),并基于该 PES 包形成段。

[0306] 对于 PG 数据的情况,如附图 37 的 B 部分所示,在将 PG 数据提供给 PID 滤波器 64 和 66 时,对 PG 数据所包括的包 $P_1, P_2, \dots$ 进行选择,并存储在缓冲器内。然后,如上所述,当缓冲器中存储的能够重构为 PES 包的 TS 包的数目足够多时,重构 PES 包。也就是说,将划分为 TS 包的段统一为 PES 包。如附图 37 的 B 部分所示,TS 包 P_1, P_2 和 P_3 形成一个段。

[0307] 对 IG 数据应用相同的处理。如附图 37 的 C 部分所示,在将 IG 数据提供给 PID 滤波器 64 和 67 时,对 IG 数据所包括的包 $I_1, I_2, \dots$ 进行选择,并存储在缓冲器内。然后,当缓冲器中存储的能够重构为 PES 包的 TS 包的数目足够多时,重构 PES 包。也就是说,将划分为 TS 包的段统一为 PES 包。如附图 37 的 C 部分所示,TS 包 I_1, I_2 和 I_3 形成一个段。

[0308] 以这样的方式,形成一个段,并在形成段的 TS 包中,当将缓冲器 60 最近输出的 TS 包提供给相应 PID 滤波器时,将其立即提供给解码器。

[0309] 对于 PG 数据,如附图 37 的 B 部分所示,在形成一个段的 TS 包 P_1, P_2 和 P_3 中,缓冲器 60 最近输出的 TS 包为 TS 包 P_1 。在将 TS 包 P_1 提供给 PID 滤波器 64 和 66 的时刻(附图 37 中的时间 RT_{p1-3}),形成一个段,并几乎在该时间 RT_{p1-3} ,将其提供给 PG 解码器 74。

[0310] 对 IG 数据应用相同的处理。如附图 37 的 C 部分所示,在形成一个段的 TS 包 I_1, I_2 和 I_3 中,缓冲器 60 最近输出的 TS 包为 TS 包 I_1 。在将 TS 包 I_1 提供给 PID 滤波器 64 和 67 的时刻(附图 37 中的时间 RT_{i1-3}),形成一个段,并几乎在该时间 RT_{i1-3} ,将其提供给 IG 解码器 77。

[0311] 因此,如果 TS 包的逻辑顺序已知,则在基于某一 TS 包的相对时间,评估要将段提供给 PG 解码器 74 或 IG 解码器 77 的时间。换句话说,基于 TS 包的顺序,能够唯一的确定将段提供给 CDB110 的时间 RT。

[0312] 如果定义存储在解码缓冲器中的 PG 数据或 IG 数据的段的数目的上限 N,则一直满足该上限 N 的数据结构将如下所述。

[0313] 现在,假定为各段指定的 DTS 的顺序与提供各段的顺序相对应。就附图 35 所示某一段 Seg_m 来说,在为早于段 Seg_m 而提供给解码缓冲器的 N 个连续的段所指定的 DTS 中,对应于最早时间的 DTS 将早于将段 Seg_m 提供给解码缓冲器的时间 RT_m 。为了满足该条件,指定各段的 DTS,并排列形成各段的 TS 包。

[0314] 如上所述,基于 TS 包的顺序,能够唯一的确定将段 Seg_m 提供给解码缓冲器的时间 RT_m 。

[0315] 以预定顺序排列包括除 PG 数据和 IG 数据以外的数据,例如,视频数据和音频数据,的 TS 包。因此,PG 数据和 IG 数据的 TS 包,和视频数据和音频数据的 TS 包相互相关。通常,最好将 PG 数据或 IG 数据的 TS 包分组到一个预定单元,诸如显示集,这样的话,能够便于管理或验证。

[0316] 以下,将根据本发明的一个实施例,给出用于产生上述数据结构的方法,和用于制造将该数据结构记录于其上的记录介质的方法的说明。附图 38 是描述用于制造一记录介质的流程图,该记录介质可以将包括视频数据,音频数据,PG 数据,和 IG 数据的多路复用流记录于其上。如附图 38 所示,从产生内容到将该内容形成为光盘的处理,被称为“创作处理”。

[0317] 利用例如,计算机上执行的预定软件,执行该创作处理。诸如视频数据,音频数据,字幕数据和菜单显示数据之类的不同材料的数据,均记录在诸如硬盘驱动器,之类的连接在计算机上的外部存储设备上。作为选择,还可以将材料数据记录在计算机的硬盘上,或者如果提供具有网络连接功能的计算机,则通过网络,从外部源提供材料数据。

[0318] 在步骤 S50 中,获得视频数据,音频数据,字幕数据和菜单显示数据,且以预定时序顺序排列上述数据项,以便确定回放时间。因此,创建将要在记录介质上记录的内容。

[0319] 然后,在步骤 S51 中,对视频数据,音频数据,字幕数据和菜单显示数据中的每一个进行编码,并从相应的数据项中产生基本码流。更具体地说,从视频数据和音频数据中产生基本码流。基于字幕数据,产生形成 PG 数据的诸如 PCS,WDS,PDS,和 ODS 的段。同样,基于菜单显示数据,产生形成 IG 数据的诸如 ICS,PDS,和 ODS 的段。对于 PG 数据和 IG 数据,均产生结束段。

[0320] 然后,在步骤 S52 中,对步骤 S51 中产生的基本码流进行多路复用。更具体地说,根据数据类型,将步骤 S51 中产生的每一个基本码流划分为较小的部分,且如果需要则将 PTS 和 DTS 添加至多个所划分的部分,且将所划分的多个部分形成一个 PES 包。然后,将 PES 包划分为较小的部分,并且将表示数据类型的 PID 添加至每一个划分后的部分,从而得到 TS 包。以对应于 DTS 的顺序,排列形成视频数据,音频数据,PG 数据和 IG 数据的 TS 包,然后,将对应数据类型的 TS 包多路复用为逻辑顺序中一个单独的数据流。

[0321] 在步骤 S53 中,执行步骤 S52 中所产生的数据流的验证。对于 PG 数据或 IG 数据,确定所产生的数据流的逻辑顺序,是否满足本发明实施例的预定数据结构的条件。如果发现该逻辑顺序满足步骤 S53 中的条件,则处理进行至步骤 S54。如果该逻辑顺序不满足该条件,则处理返回至步骤 S52。在步骤 S52 中,例如,对 TS 包进行重新排列,并重新多路复用为一个数据流,从而满足预定数据结构的条件。如果不满足步骤 S53 中的条件,则所采用的方法,并不局限于上述处理,且如附图 38 中虚线所示,根据不满足该条件的原因,该处理返回至步骤 S51,并对该数据进行重新编码。以下,将说明用于步骤 S53 中的验证步骤所指定的处理。

[0322] 在步骤 S54 中,对步骤 S53 中验证的数据流进行格式化。在格式化处理中,在步骤 S54 中,创建基于数据流,例如,上面所讨论的剪辑信息文件中存储的信息,的数据库。将该格式化处理中产生的数据和数据流,构建入一个文件系统。例如,将数据和数据流存储到多个预定文件中,然后从各文件(参见附图 6)构建一个层次结构。然后提供逻辑存取单元。与该文件系统相同,可以使用例如,统一磁盘格式(UDF)。

[0323] 此外,在 BD-ROM 标准中,将具有 4 字节数据长度的预定头部添加至 TS 包,且然后,将所得到的 TS 包作为源包记录在光盘上。以这样的方式,在步骤 S54 的格式化处理中,还可以通过将一个头部添加至 TS 包,执行源包的构成。

[0324] 然后,在步骤 S55 中,执行对所格式化数据流的验证。在该验证处理中,基于到步骤 S54 为止所产生的文件或文件系统,产生一个光盘图像,并将其回放,从而确定到步骤 S54 为止该创作处理的结果是否正确。

[0325] 产生光盘图像,作为反映目标记录介质,诸如光盘上所记录图像的数据。例如,如果对以扇区为单元的记录数据,目标记录介质是光盘,则以扇区为单元创建作为位图数据的数据。

[0326] 如果在步骤 S55 中作为验证处理的结果,确定到步骤 S54 为止的创作处理的结果是正确的,则处理进行至步骤 S56。在步骤 S56 中,从流数据中产生一个光盘。如果作为步骤 S55 中验证处理的结果,发现在所格式化数据中存在误差,即,到步骤 S54 为止的创作处理的结果是不正确的,则处理返回至对数据流重新进行格式化的步骤 S54。如果步骤 S55 中的验证处理失败,则所采用的方法,并不局限于上述处理。根据误差的原因,该处理返回至步骤 S51,并对该数据进行重新编码

[0327] 在步骤 S56 中,从步骤 S55 所验证的光盘图像中产生一个光盘。例如,对所验证的数据执行误差校正编码处理,然后,对其执行记录编码处理。然后,以预定方式对所得到的记录数据进行调制,且在现在,准备将所得到的信号记录到记录介质上。基于该信号,产生一个母盘。下面,将说明母盘产生的过程。

[0328] 在上述实例中,在多路复用处理之后执行 2 次验证处理,并且也在格式化处理之后。作为选择,仅在格式化处理之后执行 1 次验证处理。这样,如果作为验证处理的结果,发现一个误差,根据误差的类型,处理返回至步骤 S51 中的编码处理,步骤 S52 中的多路复用处理,或步骤 S54 中的格式化处理。

[0329] 下面,将特别讨论步骤 S53 中验证处理的细节。在步骤 S60 中,从记录介质上所记录的所有 TS 包中重构 PES 包,且从 PES 包中重构各段。

[0330] 然后,在步骤 S61 中确定重构的各段是否按 DTS 的顺序排列。如果在步骤 S61 中发现各段不是按 DTS 的顺序排列,则处理进行至执行误差处理的步骤 S68。例如,处理返回至附图 38 的流程图中的步骤 S52,且在多路复用处理中,重新排列 TS 包,以便使各段的顺序与 DTS 的顺序相匹配。作为选择,处理可以返回至步骤 S51,如附图 38 中虚线所示,且再一次执行编码处理,以便可以改变 DTS,并为各段提供 DTS。

[0331] 相反,如果在步骤 S61 中确定各段是按 DTS 的顺序排列,处理进行至步骤 S62。在步骤 S62 中,如果为所有的段设置 DTS,则获得各 DTS。然后,在步骤 S63 中,如上所述,基于 TS 包的顺序,确定将该段提供给解码缓冲器的时间 RT_n 。从该处理中排除不与 DTS 或 PTS 一起提供的结束段。

[0332] 然后,在步骤 S64 中,通过提供等于 N 的数量的段,将目标段的 DTS_n 与在目标段之后将段提供给解码缓冲器的时间 RT_{n+N} 相比较。如果该流程图中第一次执行步骤 S64,则目标段可以是所有段中最早时间提供给解码缓冲器的段。在步骤 S65 中,确定目标段的 DTS 是否早于时间 RT_{n+N} 。将所确定的结果存储到预定寄存器中。

[0333] 然后,在步骤 S66 中确定是否已处理除结束段之外的所有段。如果存在未处理的

段,则处理返回至步骤 S64,步骤 S64 中,将未处理的段设置为目标段,并通过提供等于 N 的数量的段,将下一个目标段的 DTS_n 与在目标段之后将段提供给解码缓冲器的时间 RT_{n+N} 相比较。

[0334] 事实上,除结束段之外的所有段中,最后处理的第 (N-1) 个段,可以从该处理中排除。

[0335] 如果步骤 S66 中所有段均已处理,则处理进行至步骤 S67,从而基于步骤 S64 中所获得的确定结果,确定是否违反解码缓冲器中能够存储的段的数目的上限 N 的条件。如果违反了上限 N 的条件,则处理进行至执行误差处理的步骤 S68。

[0336] 例如,处理返回至附图 38 的流程图中的步骤 S52。基于步骤 S67 中的确定结果,对于发现违反上限 N 的条件的一组段,即,对于形成该组的每一个段,它的 DTS_n 都晚于通过提供等于 N 的数量的段,将在目标段之后将段提供给解码缓冲器的时间 RT_{n+N} ,则对 TS 包进行重新排列和多路复用,以便使时间 RT_{n+N} 晚于 DTS_n 。作为选择,处理返回至步骤 S51,如附图 38 中虚线所示,其中,再一次执行编码处理,且改变并提供 DTS。

[0337] 在上述实例中,对将记录在记录介质上的所有段,执行验证处理。然而,可以在每一个播放列表的一组段上执行验证处理。作为选择,可以对每一个标题的一组段执行验证处理。

[0338] 附图 39 中示出的处理仅仅表示实现本发明一个实施例的一种方法的一个实例,且本发明并不局限于该处理。

[0339] 附图 39 中示出的处理是附图 38 所讨论的创作处理的一个步骤。然而,附图 39 所示出的处理可以作为用于所创建光盘或完整的光盘的验证处理而独立执行。这样,在步骤 S68 的误差处理中,可能指出误差的存在,且如果存在的话,还指出误差的位置。

[0340] 附图 40 描述了执行附图 38 和 39 所讨论的创作处理的创作设备 2 的配置。该创作设备 2 包括 CPU, RAM, ROM, 硬盘驱动器等等,虽然它们并未示出。CPU 通过使用 RAM 作为网络存储器,基于 ROM 或硬盘驱动器中存储的程序,控制整个创作设备 2。利用,例如,计算机,形成该创作设备 2。作为选择,可以通过用于执行创作处理的专用硬件形成该创作设备 2。

[0341] 存储器 200 包括一个或多个硬盘驱动器,并将所要记录的内容材料,诸如视频数据,音频数据,字幕数据,和用于显示菜单屏幕的按钮图像数据,存储在光盘上。

[0342] 用户界面 (UI) 单元 204 提供用户界面,该用户界面对存储器 200 所存储的数据进行选择或创建,该数据诸如作为内容材料的视频数据或音频数据,与内容材料一起显示的字幕数据,和用于显示菜单屏幕的按钮图像数据。UI 单元 204 在监视器 (未示出) 上显示用户界面屏幕,诸如附图 41 所示。该用户基于用户界面屏幕,在创作设备 2 上执行预定操作,从而允许创作设备 2 执行创作处理。

[0343] 在附图 41 所示实例中,用户界面屏幕包括时间轴屏幕 400 和存储屏幕 410。在存储屏幕 410 上,显示存储器 200 中存储的数据列表。在存储屏幕 410 上,例如,内容材料 411,诸如视频数据和音频数据,将被显示在区域 410A 中,在区域 410B 中存储字幕图像数据 412,且在区域 410C 中存储用于显示菜单屏幕的,诸如按钮图像的图像数据 413,将上述数据作为预定图标显示。

[0344] UI 单元 204 还显示时间轴屏幕 400。在时间轴屏幕 400 上,水平轴表示时间序列,

且在时间比例尺 401 上示出时间信息。用户选择通过使用指定装置,诸如鼠标,用已知方法,诸如拖放操作,在存储屏幕 410 上显示的内容材料,字幕图像数据,或图像数据,诸如按钮图像,并且基于时间比例尺 401 将所选定的数据放置在时间轴屏幕 400 上。在附图 41 所示的实例中,在时间轴屏幕 400 上安排视频和音频数据 402,字幕图像数据 403A 和 403B,以及菜单屏幕 404。基于数据的位置,确定单个数据的显示时间和流结构。

[0345] 附图 41 所示的用户界面屏幕,是仅仅用于实现该实施例的实例,且该实施例中使用的用户界面屏幕并不局限于附图 41 中所示出的。

[0346] 响应于来自于 UI 单元 204 的开始对数据进行编码的指令,根据用户执行的预定操作,基于对时间轴屏幕 400 所显示数据的选择,从存储器 200 读取视频数据,音频数据,字幕图像数据,和图像数据,诸如显示菜单屏幕所用的按钮图像,且将上述数据提供给视频编码器 201,音频编码器 202 和数据编码器 203。

[0347] 视频编码器 201 和音频编码器 202 分别对从存储器 200 中读取的视频数据和音频数据进行编码,从而分别产生视频数据基本码流和音频数据基本码流。

[0348] 数据编码器 203 对从存储器 200 中读取的字幕图像数据进行编码,从而产生由 PG 数据,即,PG 流,表示的基本码流。数据编码器 203 也对从存储器 200 中读取的图像数据,诸如显示菜单屏幕所用的按钮图像数据,进行编码,从而产生由 IG 数据,即,IG 流,表示的基本码流。

[0349] 将从视频编码器 201,音频编码器 202 和数据编码器 203 输出的数据,提供给多路复用器 (MUX) 205。多路复用器 205 将每个数据划分为多个预定部分,且如果需要,将 DTS 和 / 或 PTS 添加至每一个划分后的部分,从而构建 PES 包。多路复用器 205 进一步将 PES 包划分为多个预定部分,且将表示数据类型 的 PID 添加至每一个划分后的部分,从而形成 TS 包。然后,以预定顺序排列 TS 包。因此,产生将视频数据基本码流,音频数据基本码流,PG 流和 IG 流进行多路复用的传输流。

[0350] 将从多路复用器 205 输出的传输流,提供给格式化单元 206,且将具有 4 字节数据长度的预定头部,添加至每一个 TS 包,以便将 TS 包转换为源包。格式化单元 206 还基于所提供的传输流,产生诸如剪辑信息文件的数据库。将所产生的数据和数据流,构建入预定文件系统,诸如 UDF。

[0351] 格式化单元 206 还基于所产生的数据,产生光盘图像 208。将光盘图像 208 记录在硬盘驱动器 (未示出) 上。

[0352] 将格式化单元 206 的输出,提供给验证单元 207。验证单元 207 执行附图 38 的步骤 S53 中执行的验证处理,即,附图 39 的流程图中执行的验证,诸如段的顺序的验证,和对是否满足回放设备的解码缓冲器所存储的段的数目的上限 N 的条件的验证。验证单元 207 还执行附图 38 的步骤 S55 中执行的验证,即,对格式化数据流的验证。这样,对格式化单元 206 产生的光盘图像 208 执行验证。

[0353] 如果作为验证单元 207 所执行的验证处理的结果,发现一个误差,根据误差的类型,以预定方式控制视频编码器 201,音频编码器 202,数据编码器 203,多路复用器 205,和 / 或格式化单元 207,并重新执行相应处理。

[0354] 如果以段的排列顺序发现一个误差,或如果不满足回放设备的解码缓冲器所存储的段的数目的上限 N 的条件,则控制视频编码器 201,音频编码器 202,和数据编码器 203,从

而再次设置 DTS,或控制多路复用器 205 从而改变 TS 包的顺序。如果在数据库,源包,或文件系统中发现误差,则控制格式化单元 206,从而再次格式化数据。

[0355] 上述方法仅仅是处理误差的实例。作为选择,如果以段的排列顺序发现一个误差,或如果不满足回放设备的解码缓冲器所存储的段的数目的上限 N 的条件,则可以将表示误差的消息提供给 UI 单元 204,且用户可以手工改变包的顺序。

[0356] 如果作为验证单元 207 执行的验证处理的结果,并未发现误差,则将表示光盘图像 208 的数据从验证单元 207 输出至下面将讨论的母盘前期处理。

[0357] 上述实例中,验证单元 207 是创作设备 2 的一部分。然而,验证单元 207 可以作为独立验证装置的功能。例如,可以构建设没有验证单元 207 的创作设备 2,如附图 42 所示。在格式化单元 206 中产生光盘图像 208。在将光盘图像 208 提供给母盘前期处理器之前,将其提供给执行数据结构的验证的,独立的验证装置 3。如果作为验证的结果,发现一个误差,如上所述,则显示一个误差信息,且如果可能,还显示误差的原因和位置,例如在验证装置 3 所提供的显示单元 3A 上。如果创作设备 2' 是误差的原因,则验证装置 3 控制创作设备 2',如上所述,从而再次执行编码处理或多路复用处理。

[0358] 可以利用计算机,诸如个人计算机,形成上述验证装置 3。这样,通过该计算机所安装的软件,实现验证装置 3 所执行的验证处理。作为选择,可以利用用于执行验证处理的专用硬件形成验证装置 3。

[0359] 上述实例中,仅仅在格式化单元 206 之后的阶段提供验证单元 207。然而,例如,在多路复用器 205 和格式化单元 206 之间,放置另一个验证单元。这样,验证单元 207 可以仅为格式化单元 206 所执行的执行验证。在多路复用器 205 和格式化单元 206 之间放置的验证单元中,执行对段的顺序的验证,或对于是否满足回放设备的解码缓冲器所存储的段的数目的上限 N 的条件的验证。

[0360] 附图 43 描述了基于附图 40 或附图 42 所示创作设备 2 或 2' 输出的数据,用于制造可由回放设备 1 重写的光盘的制造方法。

[0361] 如附图 43 中所示,准备了例如玻璃的源盘。然后将诸如光阻材料之类的记录材料应用于该源盘。然后制造一个记录母盘。

[0362] 如果需要,对创作设备 2 或 2' 输出的数据执行编辑(母盘前期处理),以便产生具有将要记录在光盘上的格式的记录信号。然后,根据该记录信号调制激光束,并将其应用至母盘上的光阻材料。然后,将母盘上的光阻材料暴露给反映该记录信号的激光束。

[0363] 随后,对母盘进行显影,以便在母盘上形成凹槽。然后,对母盘执行预定处理,诸如电铸,以便制作具有母盘所传送的凹槽的金属母盘。然后,从金属母盘中制作作为模子的金属压模。

[0364] 然后,将一种材料,诸如丙烯酸纤维(PMMA)或聚碳酸酯(PC),注入模子,以便固定该模子。作为选择,该金属压模可以用紫外线(UV)固化树脂(2P)镀膜,然后,该树脂已具有 UV 辐射且已硬化。所以,将金属压模上的凹槽传送至该树脂,以便形成树脂拷贝。

[0365] 然后,在拷贝上通过汽相淀积,溅射,或旋涂法形成反射膜。

[0366] 此后,在执行进一步处理,诸如对所得到的光盘的内部和外部直径的调整,之后,切分两个光盘。在将标签和插孔添加到所得到的光盘上之后,将该光盘插入光盘盒。以这样的方式,完成将可由回放设备 1 重写的数据记录在其上的光盘。

[0367] 上述实例中,基于创作设备 2 的输出,制作母盘。然而可基于创作设备 2 所产生的光盘图像 208,将数据记录在诸如蓝光光盘之类的记录介质上。

[0368] 本领域技术人员应当理解的是,根据设计需要或其他因素,可以在本发明所附的权利要求或其等价物的范围之内,做出各种修改、组合、次组合和替换。

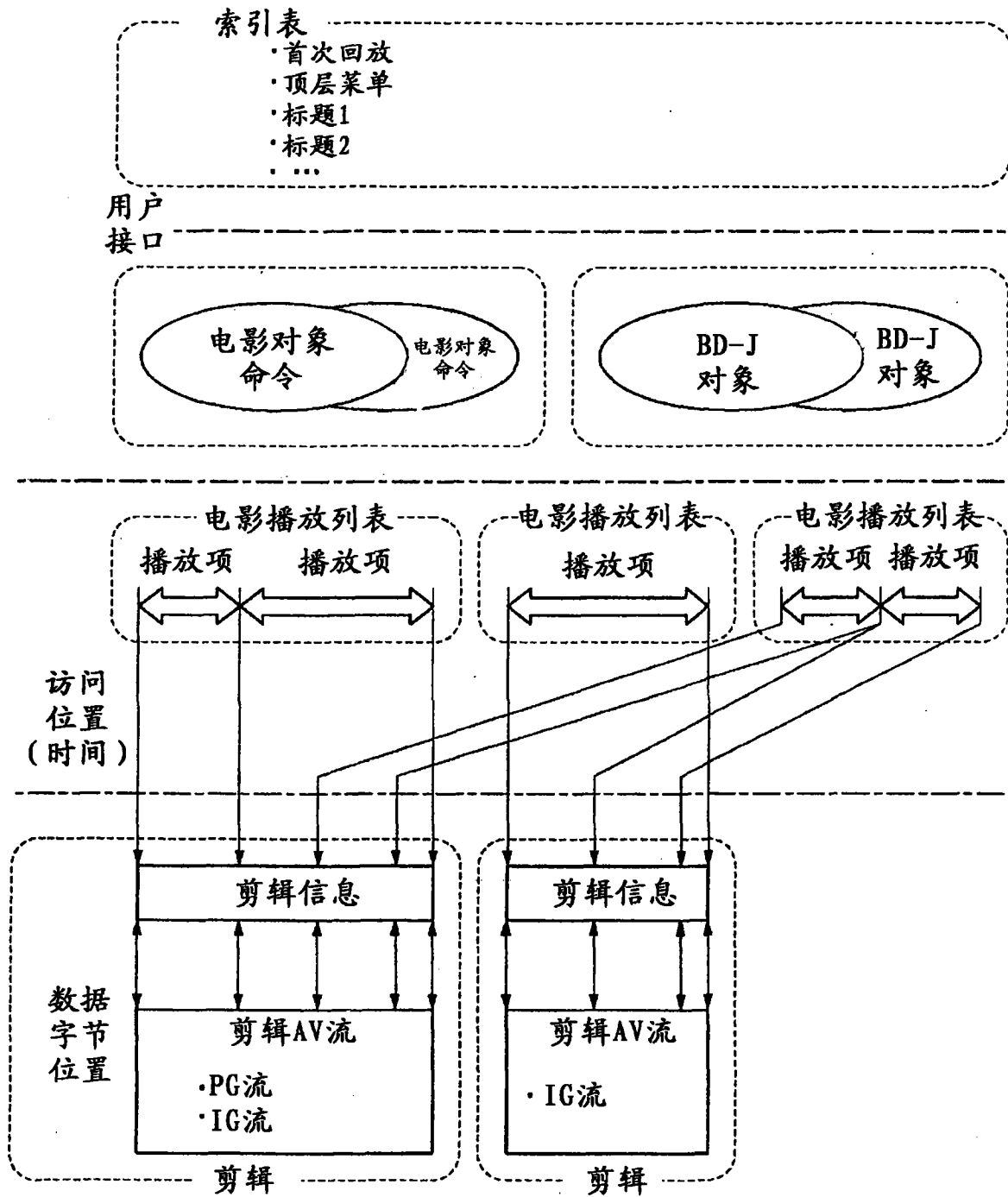


图1

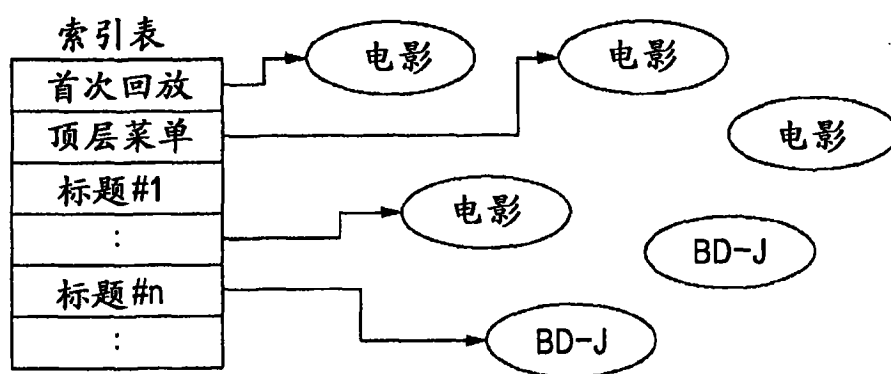


图 2

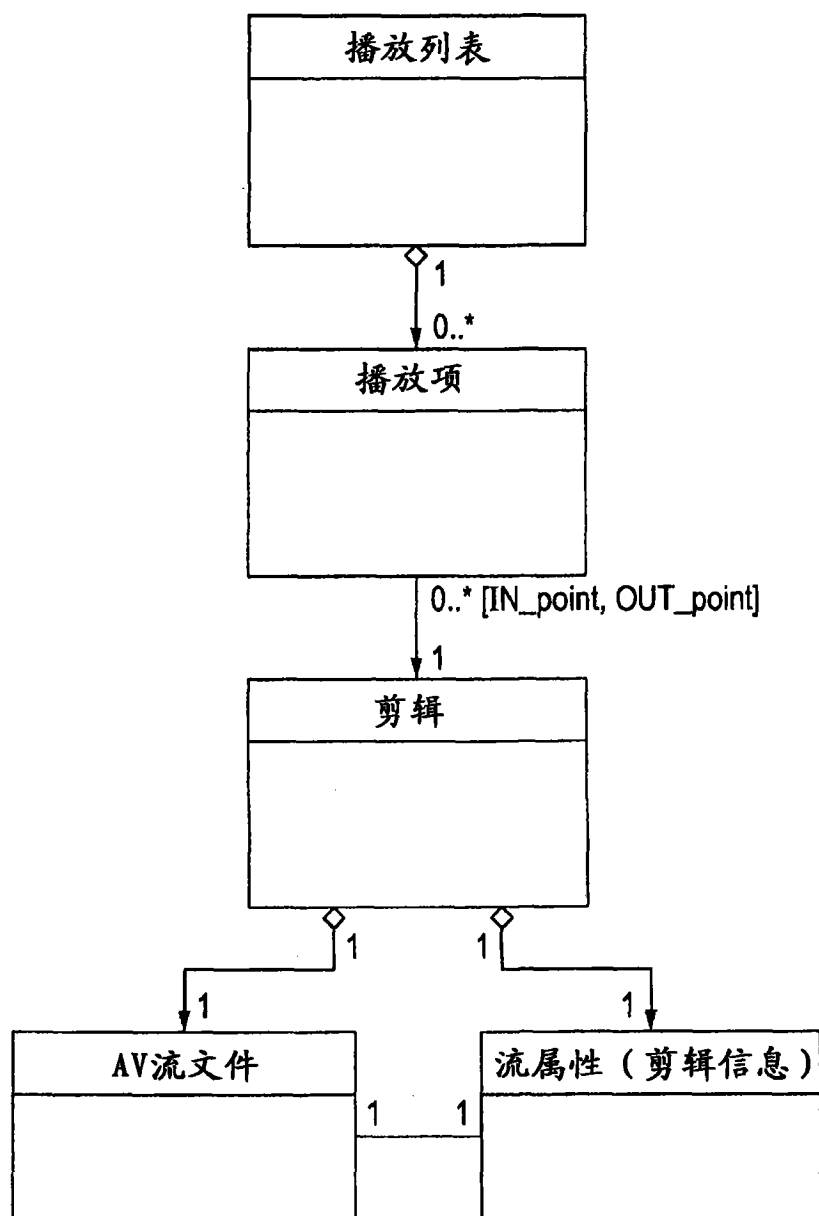


图 3

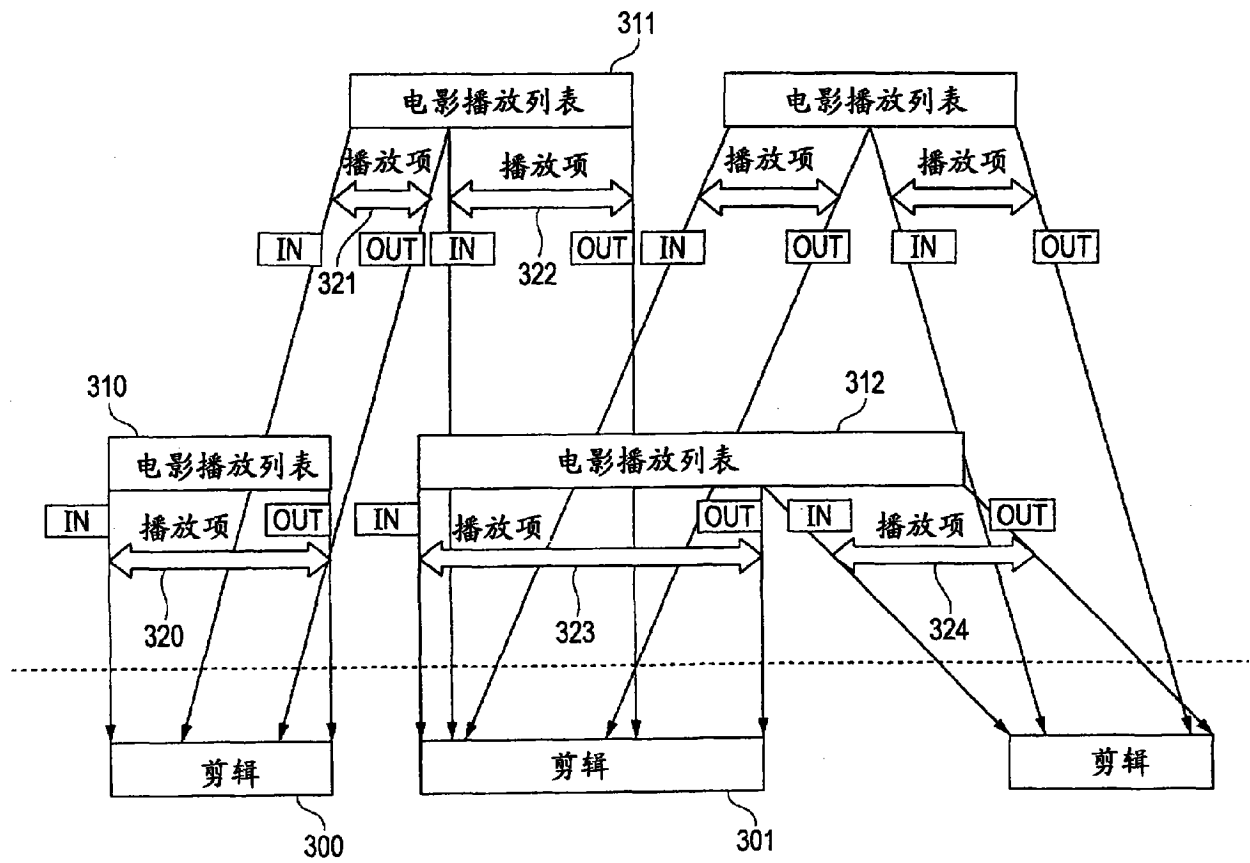


图 4

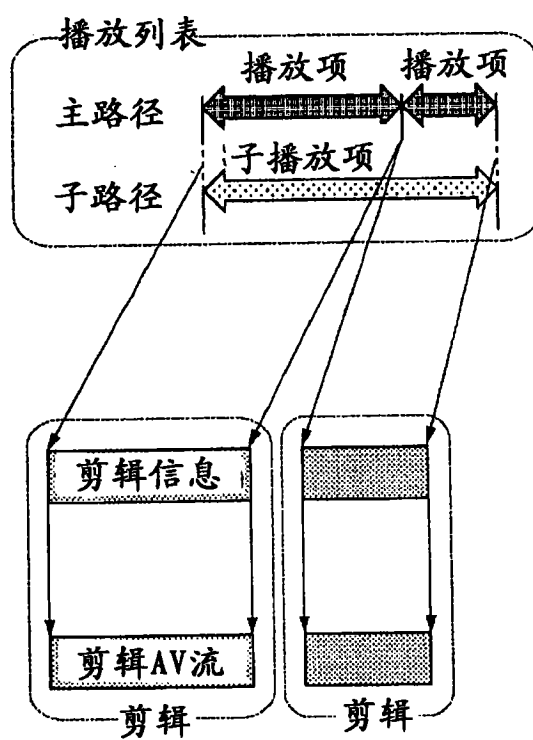


图5

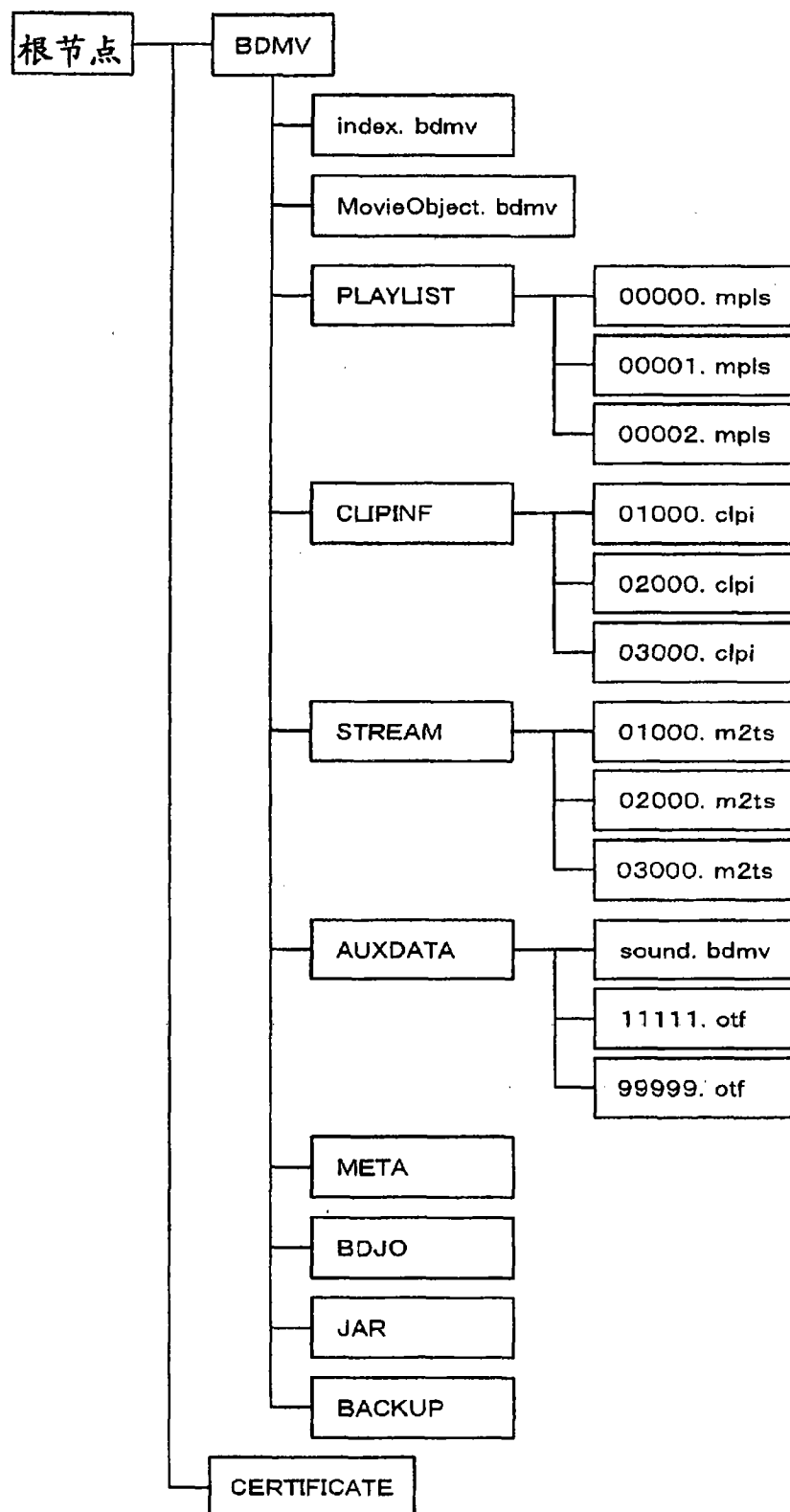


图 6

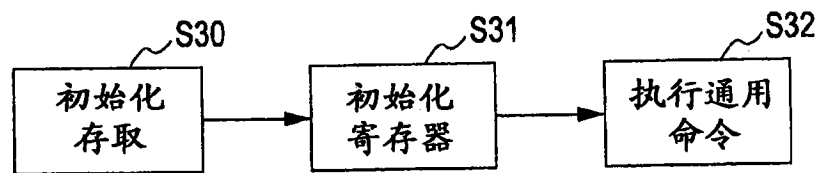


图 7A

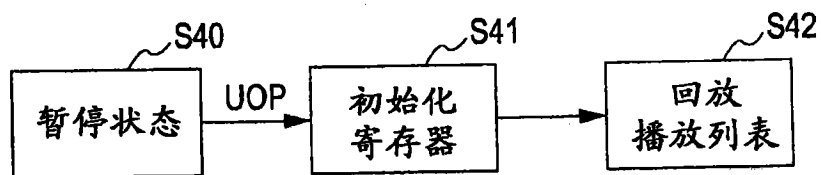


图 7B

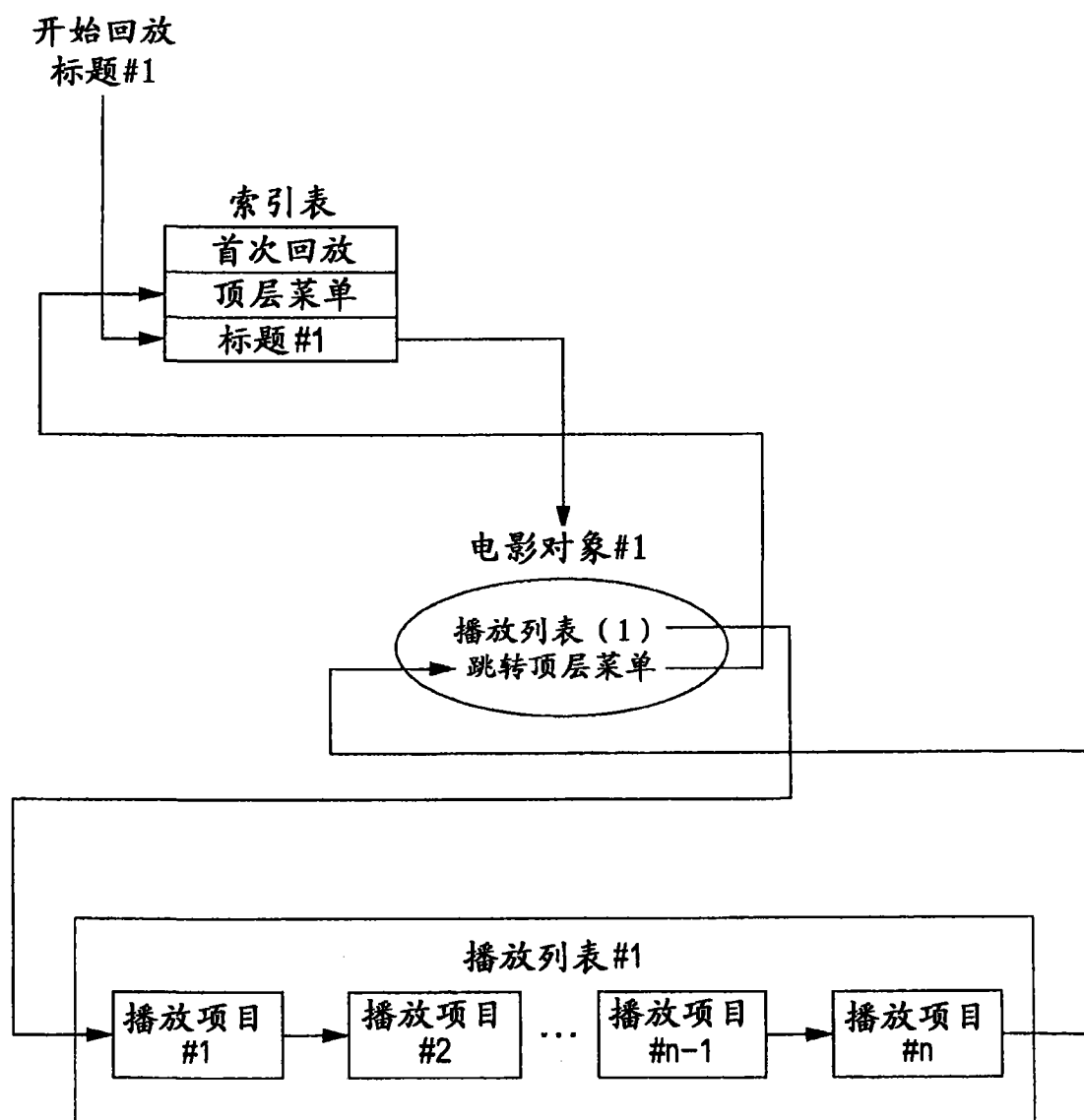


图 8

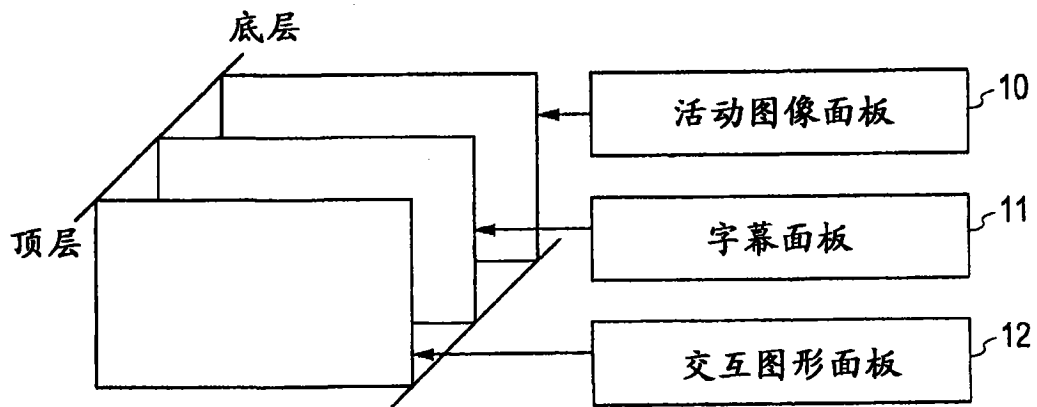


图9

项目	定义
活动图像面板	1920×1080×16 位, YCbCr (4:2:2) 每8位
字幕面板	1920×1080×8 位, 8位颜色映射地址 (调色板) + 256 α 混合
交互图形面板	1920×1080×8 位, 8位颜色映射地址 (调色板) + 256 α 混合

图10

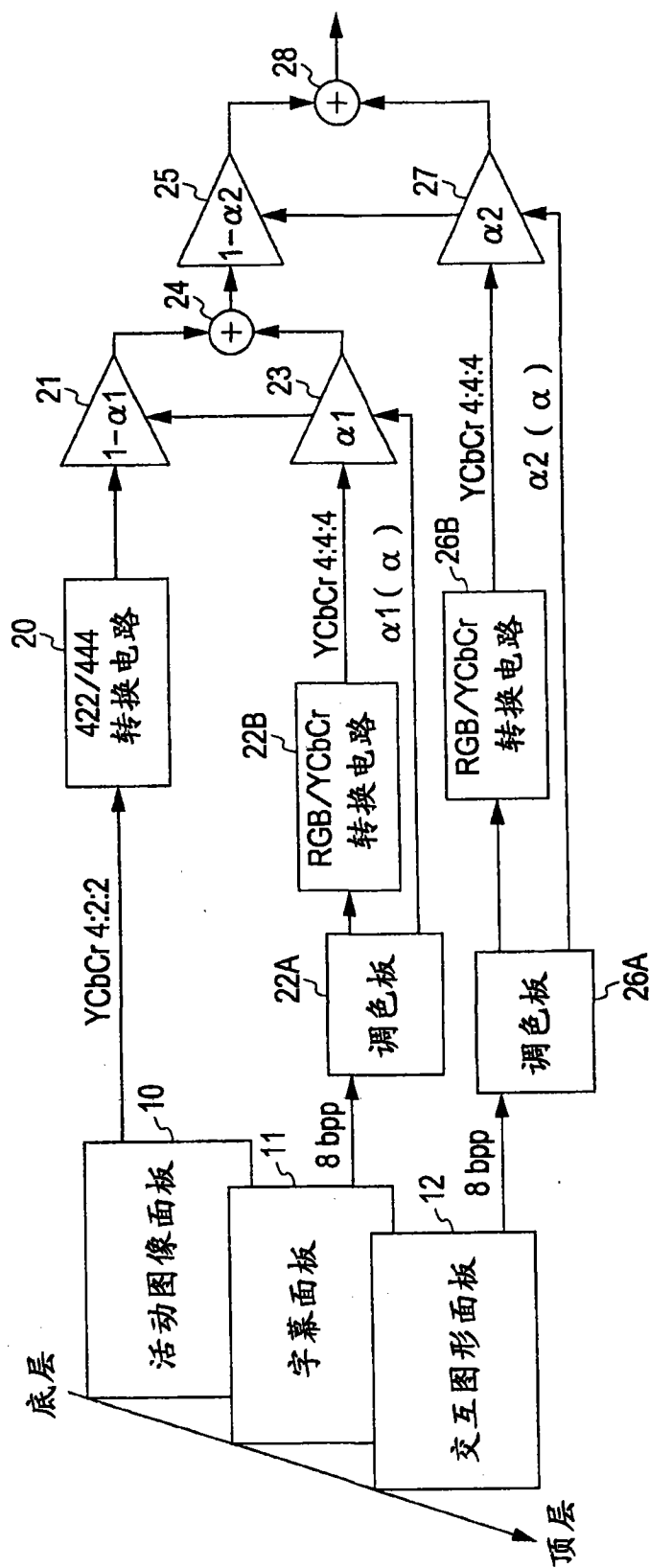


图11

输入	输入地址 8位
输出	输出数据 8位×4 (R, G, B, α)输出

图12

	三个主要颜色			不透明性
颜色索引值	R	G	B	α
0x00	0	0	0	0
0x01	10	100	30	0.5
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
0xFF	200	255	100	0.8

图13

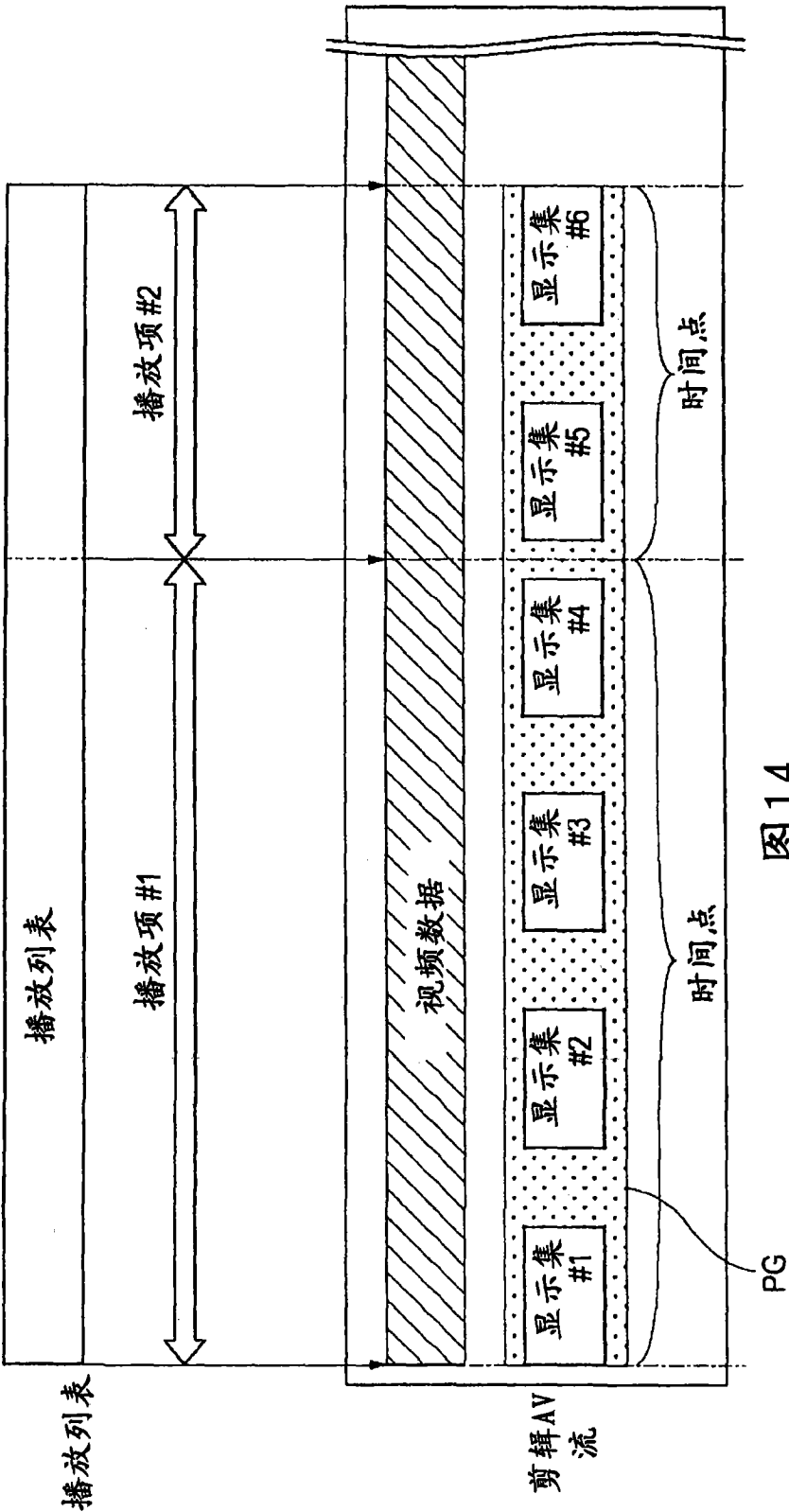


图 14

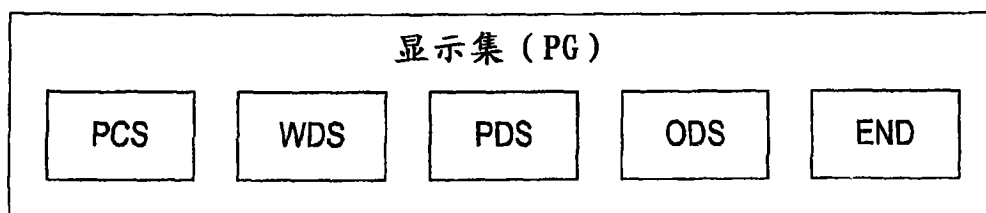


图15A

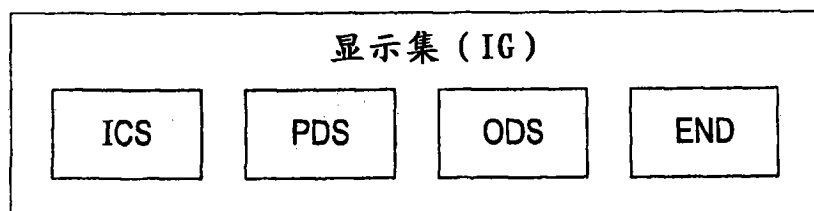


图15B

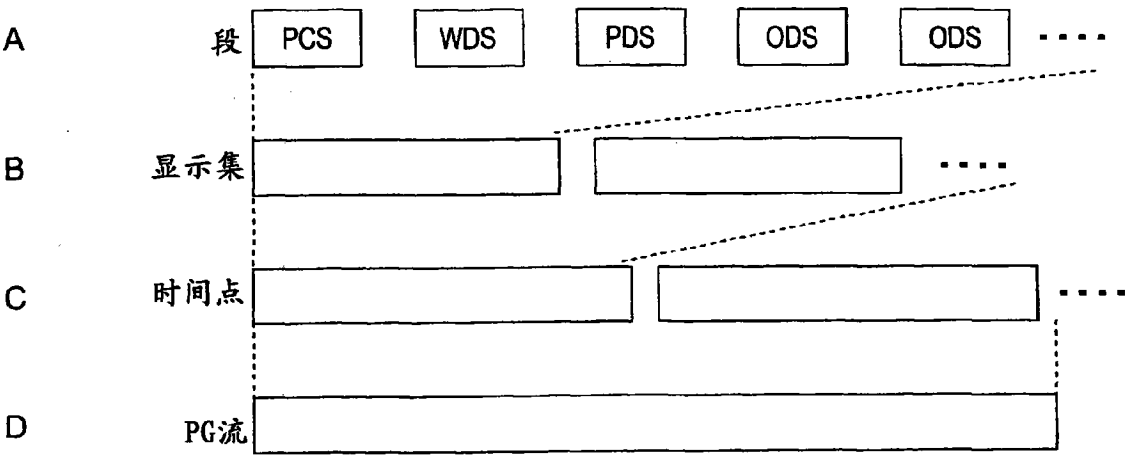


图16

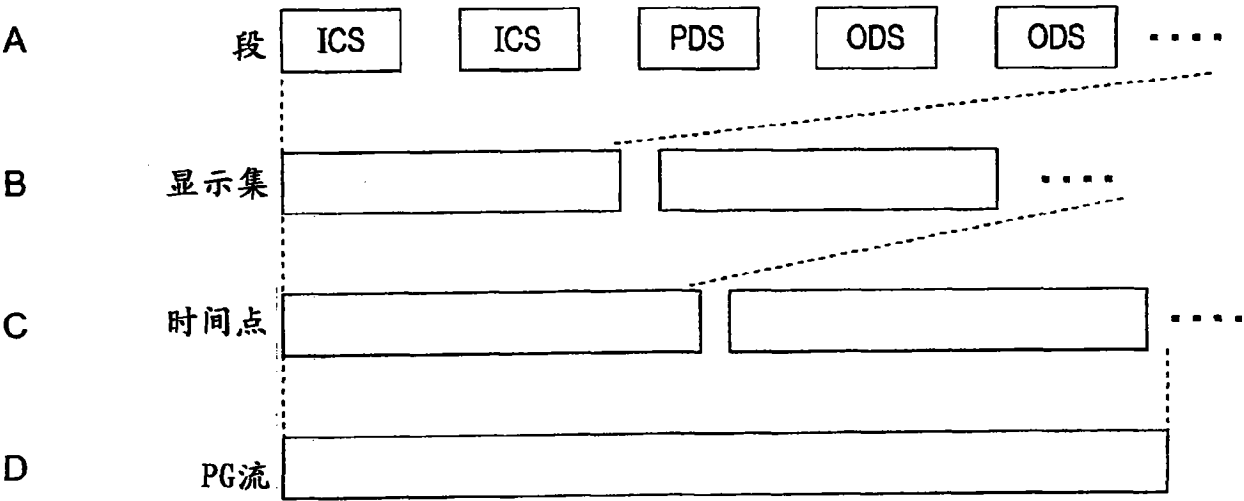


图17

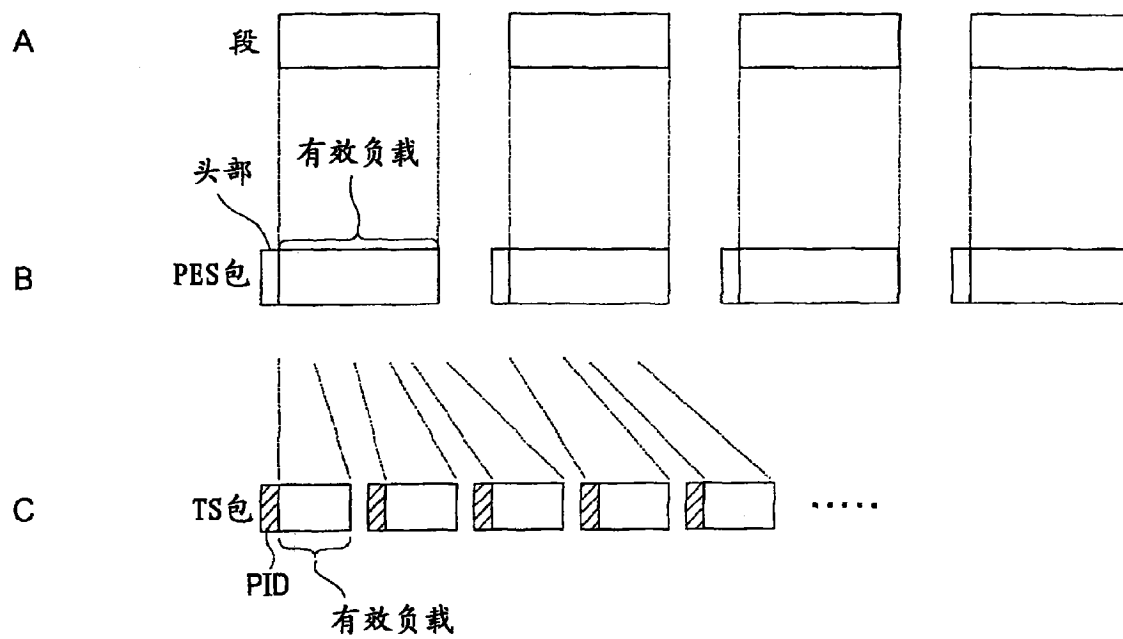


图18

语法	数据长度 (位)	助记符
presentation_composition_segment() {		
segment_descriptor()		
video_descriptor()		
composition_descriptor()		
palette_update_flag	1	bslbf
reserved	7	bslbf
palette_id_ref	8	uimsbf
number_of_composition_objects	8	uimsbf
for (i=0; i<number_of_composition_objects; i++) {		
composition_object()		
}		
}		

图19

语法	数据长度 (位)	助记符
composition_object() {		
object_id_ref	16	uimsbf
window_id_ref	8	uimsbf
object_cropped_flag	1	bslbf
forced_on_flag	1	bslbf
Reserved	6	bslbf
composition_object_horizontal_position	16	uimsbf
composition_object_vertical_position	16	uimsbf
if (object_cropped_flag==1 _b) {		
cropping_rectangle() {		
object_cropping_horizontal_position	16	uimsbf
object_cropping_vertical_position	16	uimsbf
object_cropping_width	16	uimsbf
object_cropping_height	16	uimsbf
}		
}		
}		

图20

语法	数据长度 (位)	助记符
window_definition_segment() {		
segment_descriptor()		
number_of_windows	8	uimsbf
for (i=0; i<number_of_windows; i++) {		
window()		
}		
}		

图 21

语法	数据长度 (位)	助记符
window() {		
window_id	8	uimsbf
window_horizontal_position	16	uimsbf
window_vertical_position	16	uimsbf
window_width	16	uimsbf
window_height	16	uimsbf
}		

图 22

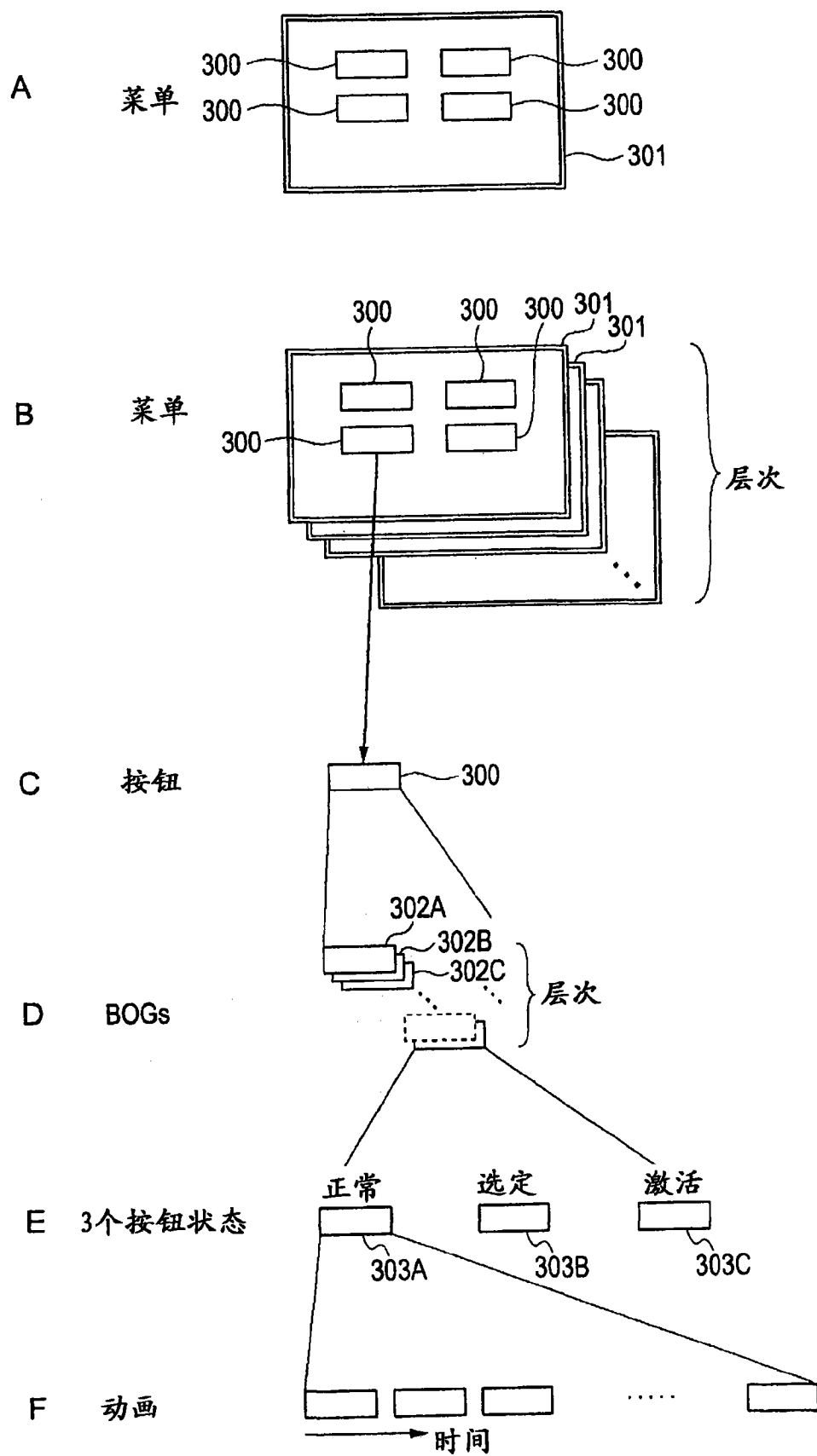


图 23

语法	数据长度 (位)	助记符
interactive_composition_segment() {		
segment_descriptor()		
video_descriptor()		
composition_descriptor()		
sequence_descriptor()		
interactive_composition_data_fragment()		
}		

图 24

语法	数据长度 (位)	助记符
interactive_composition() {		
interactive_composition_length	24	uimsbf
stream_model	1	bslbf
user_interface_model	1	bslbf
Reserved	6	bslbf
If (stream_model == '0 _b ') {		
Reserved	7	bslbf
composition_time_out_pts	33	uimsbf
Reserved	7	bslbf
selection_time_out_pts	33	uimsbf
}		
user_time_out_duration	24	uimsbf
number_of_pages	8	uimsbf
for (i=0; i<number_of_pages; i++) {		
page()		
}		
}		

图 25

语法	数据长度 (位)	助记符
page() {		
page_id	8	uimsbf
page_version_number	8	uimsbf
UO_mask_table()		
in_effects() {		
effect_sequence()		
}		
out_effects() {		
effect_sequence()		
}		
animation_frame_rate_code	8	uimsbf
default_selected_button_id_ref	16	uimsbf
default_activated_button_id_ref	16	uimsbf
palette_id_ref	8	uimsbf
number_of_BOGs	8	uimsbf
for (i=0; i<number_of_BOGs; i++) {		
button_overlap_group()		
}		
}		

图 26

语法	数据长度 (位)	助记符
button_overlap_group() {		
default_valid_button_id_ref	16	uimsbf
number_of_buttons	8	uimsbf
for (i=0; i<number_of_buttons; i++) {		
button()		
}		
}		

图 27

语法	数据长度 (位)	助记符
button() {		
button_id	16	uimsbf
button_numeric_select_value	16	uimsbf
auto_action_flag	1	bslbf
reserved	7	bslbf
button_horizontal_position	16	uimsbf
button_vertical_position	16	uimsbf
neighbor_info() {		
upper_button_id_ref	16	uimsbf
lower_button_id_ref	16	uimsbf
left_button_id_ref	16	uimsbf
right_button_id_ref	16	uimsbf
}		
normal_state_info() {		
normal_start_object_id_ref	16	uimsbf
normal_end_object_id_ref	16	uimsbf
normal_repeat_flag	1	bslbf
normal_complete_flag	1	bslbf
reserved	6	bslbf
}		
selected_state_info() {		
selected_state_sound_id_ref	8	uimsbf
selected_start_object_id_ref	16	uimsbf
selected_end_object_id_ref	16	uimsbf
selected_repeat_flag	1	bslbf
selected_complete_flag	1	bslbf
reserved	6	bslbf
}		
activated_state_info() {		
activated_state_sound_id_ref	8	uimsbf
activated_start_object_id_ref	16	uimsbf
activated_end_object_id_ref	16	uimsbf
}		
number_of_navigation_commands	16	uimsbf
for (i=0; i<number_of_navigation_commands; i++) {		
navigation_command()		
}		
}		

图 28

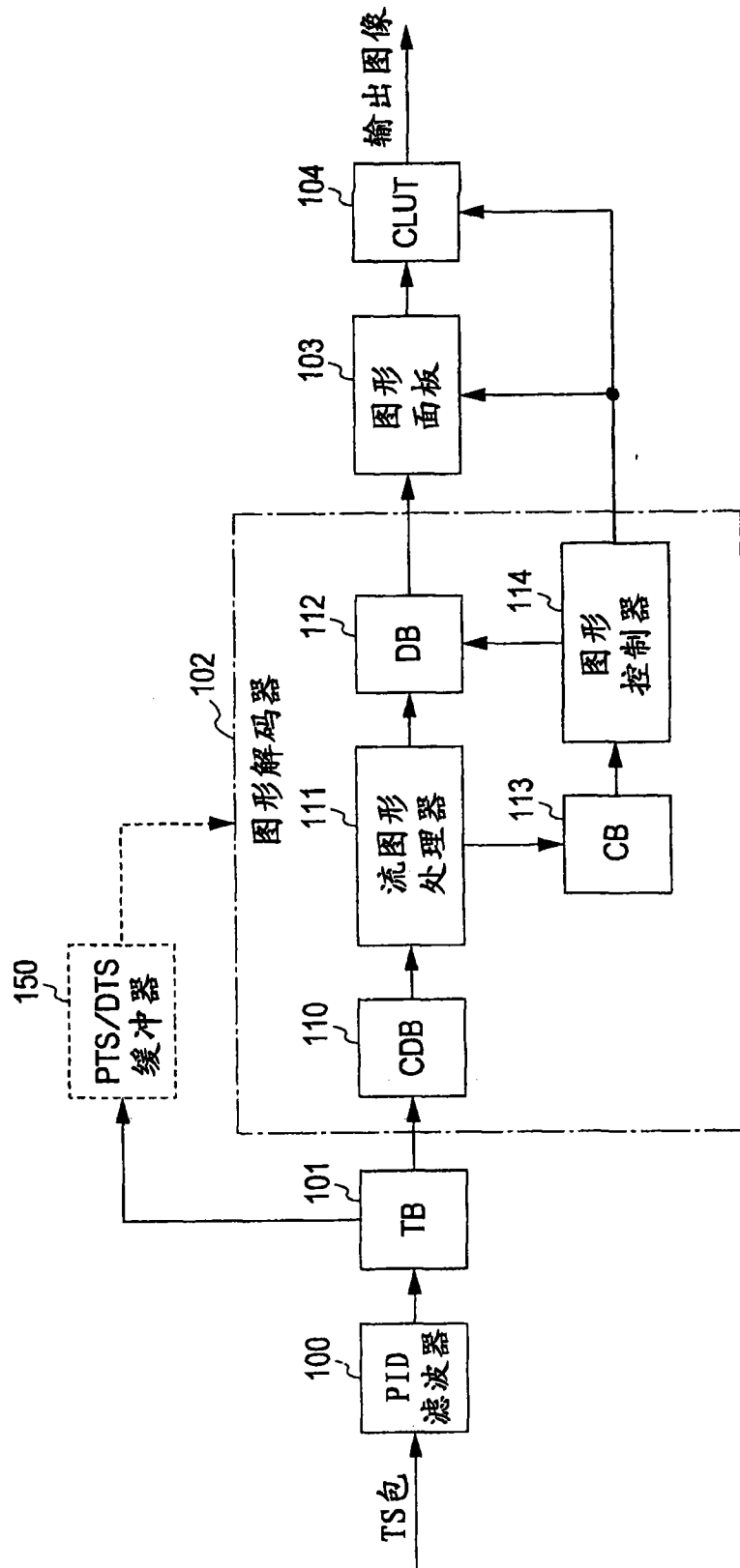


图29

段ID	DTS	PTS	(版本)	(段类型)
(16位)	33位	33位	(8位)	(4位)

PCS, ICS, PDS, WDS, ODS

图 30

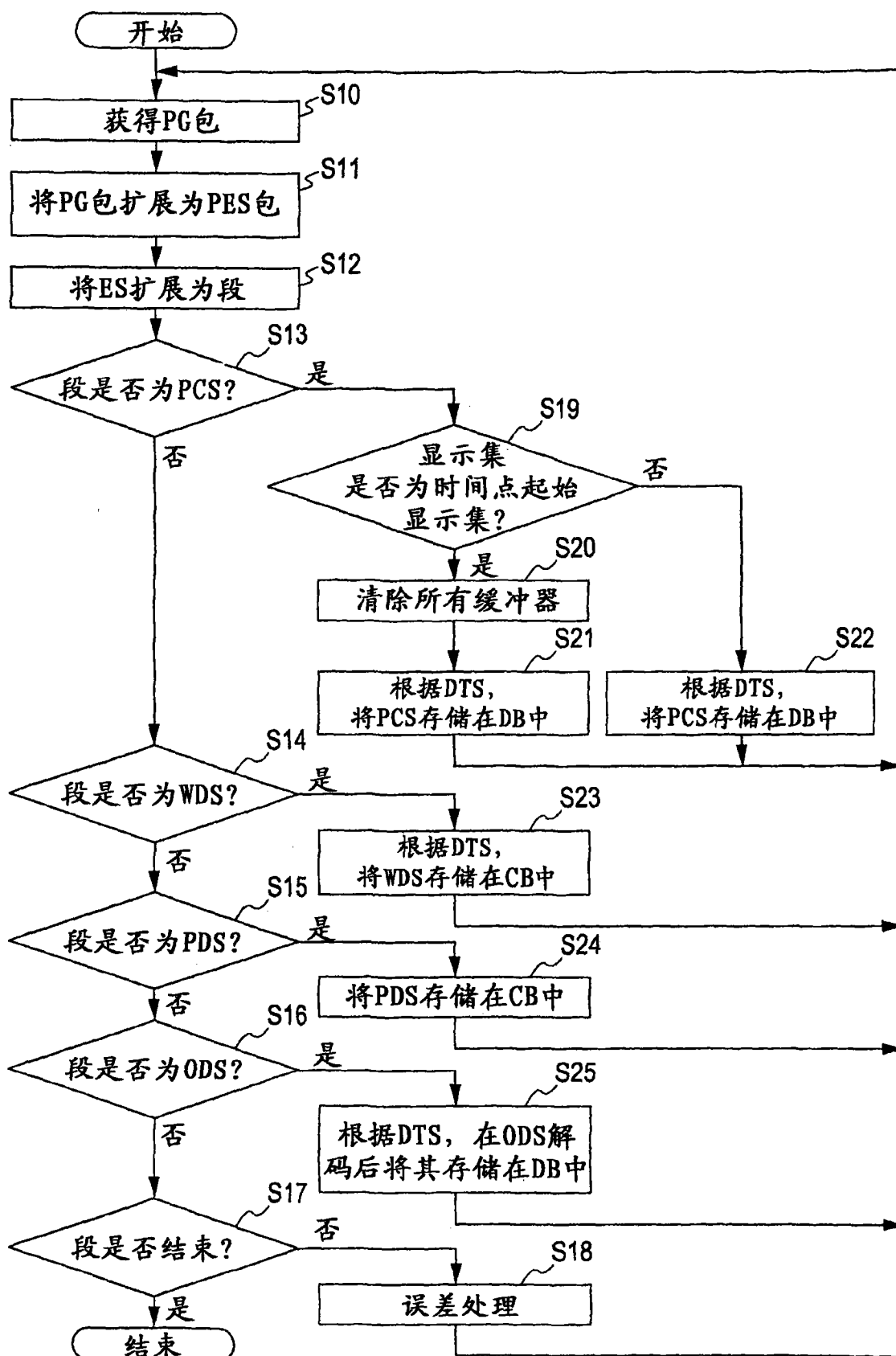


图 31

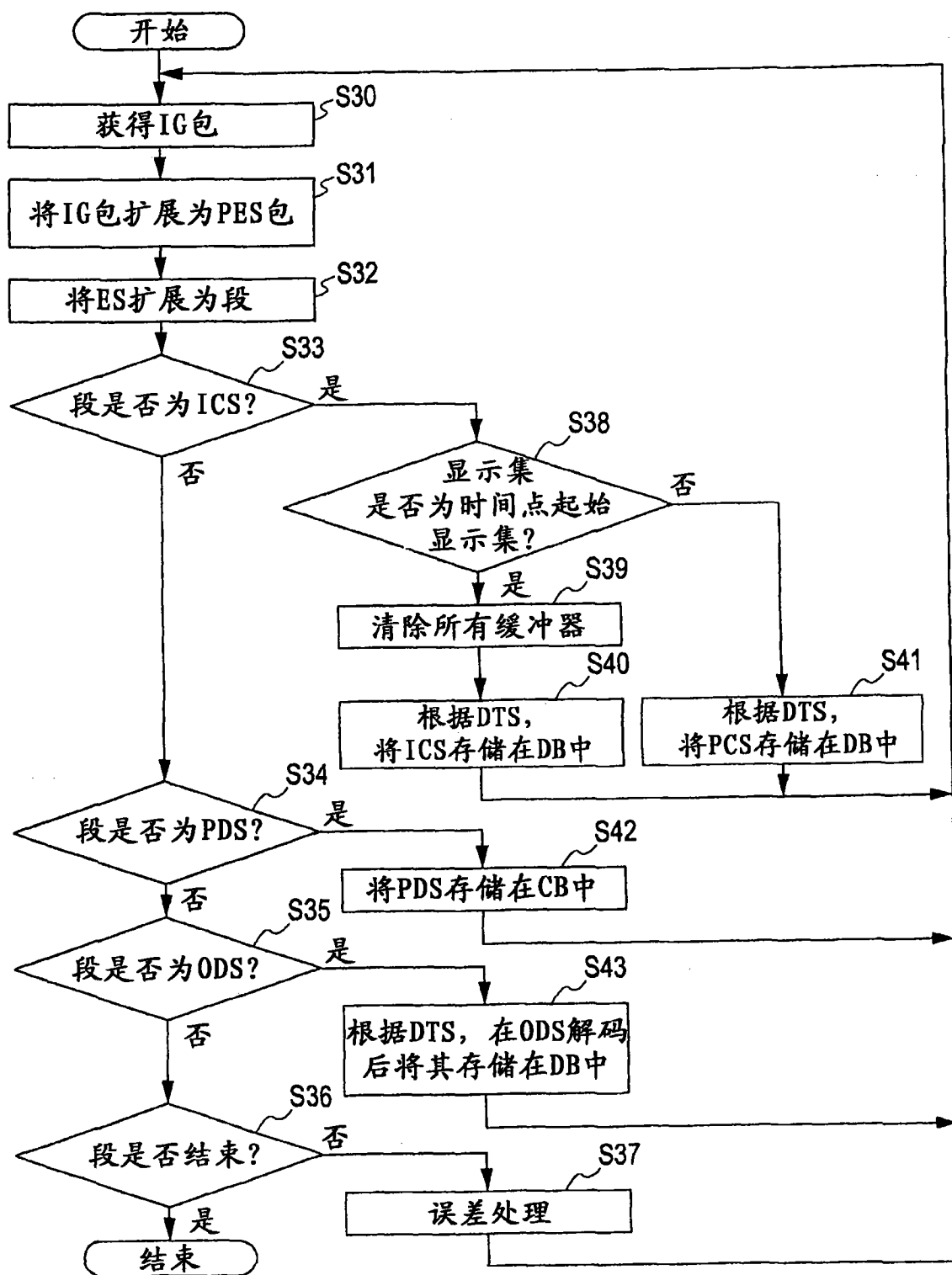
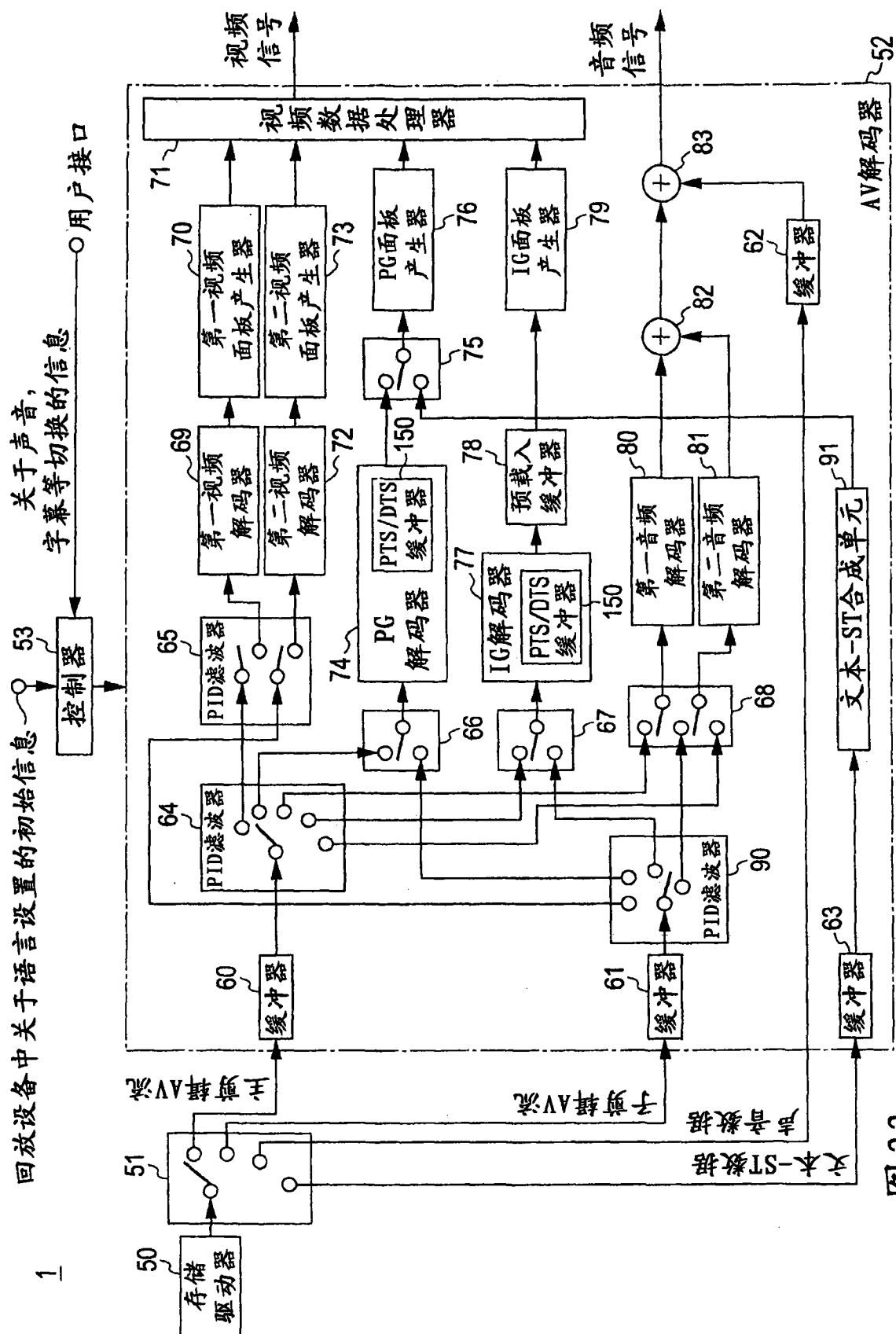


图 32



PG

段	上限 (段的数目)
PCS	8
WDS	1
PDS	8
ODS	64
END	8

图 34A

IG

段	上限 (段的数目)
ICS	1
PDS	256
ODS	4096
END	1

图 34B

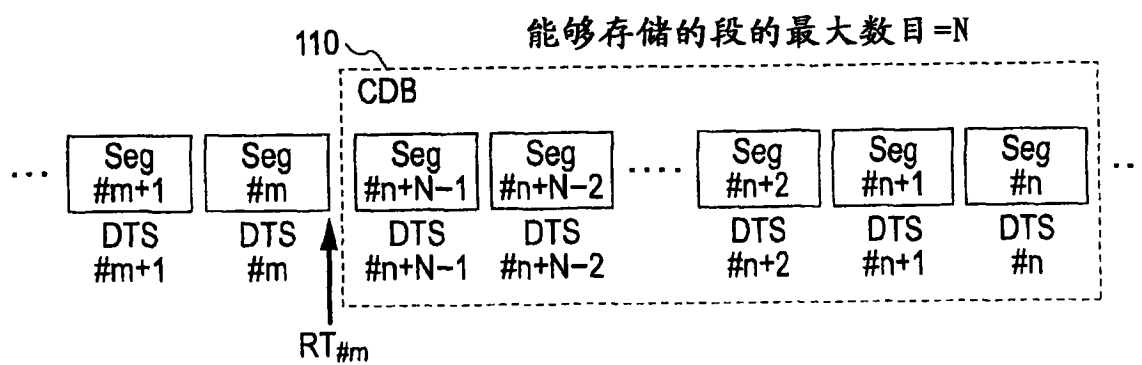


图 35

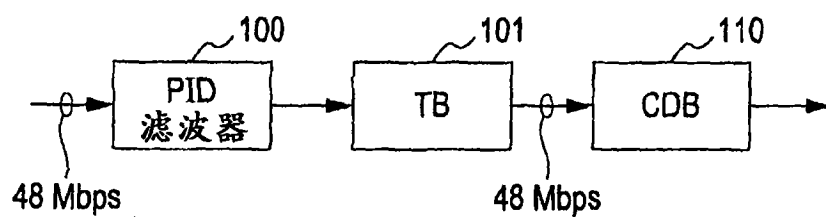


图 36A

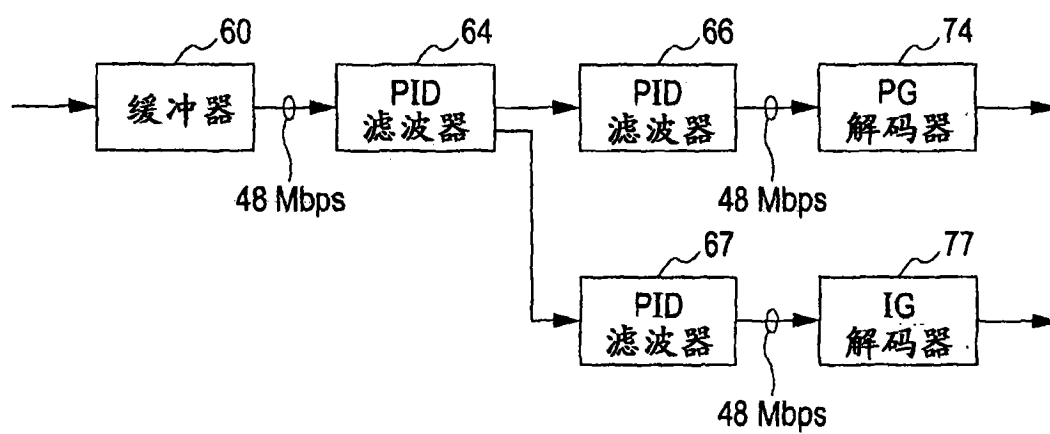


图 36B

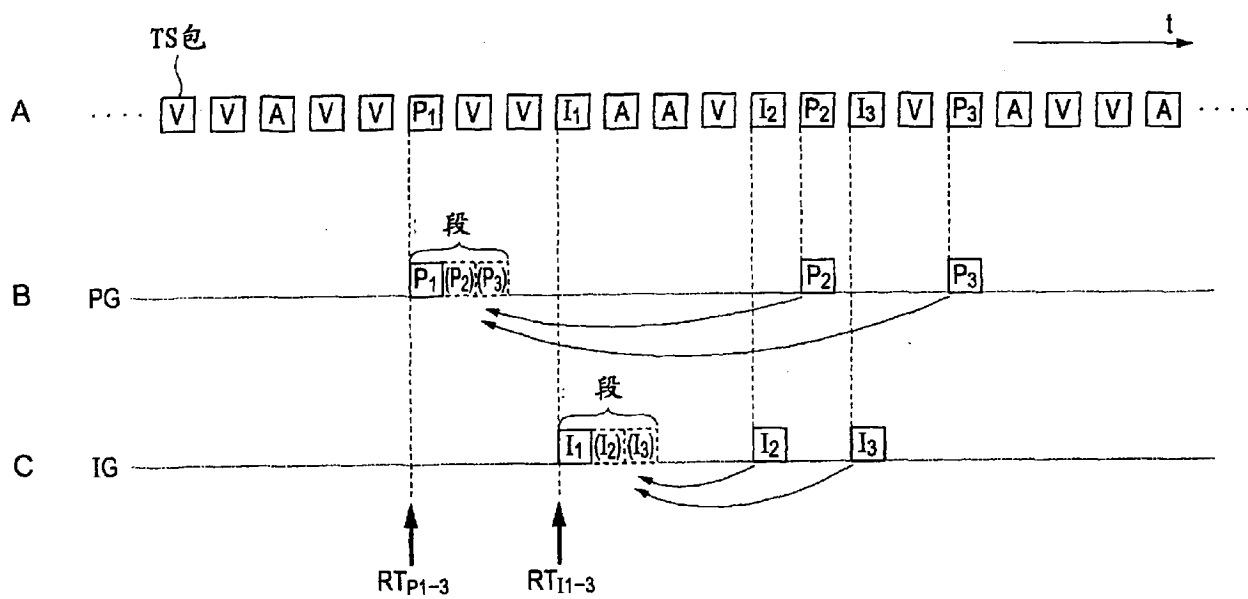


图 37

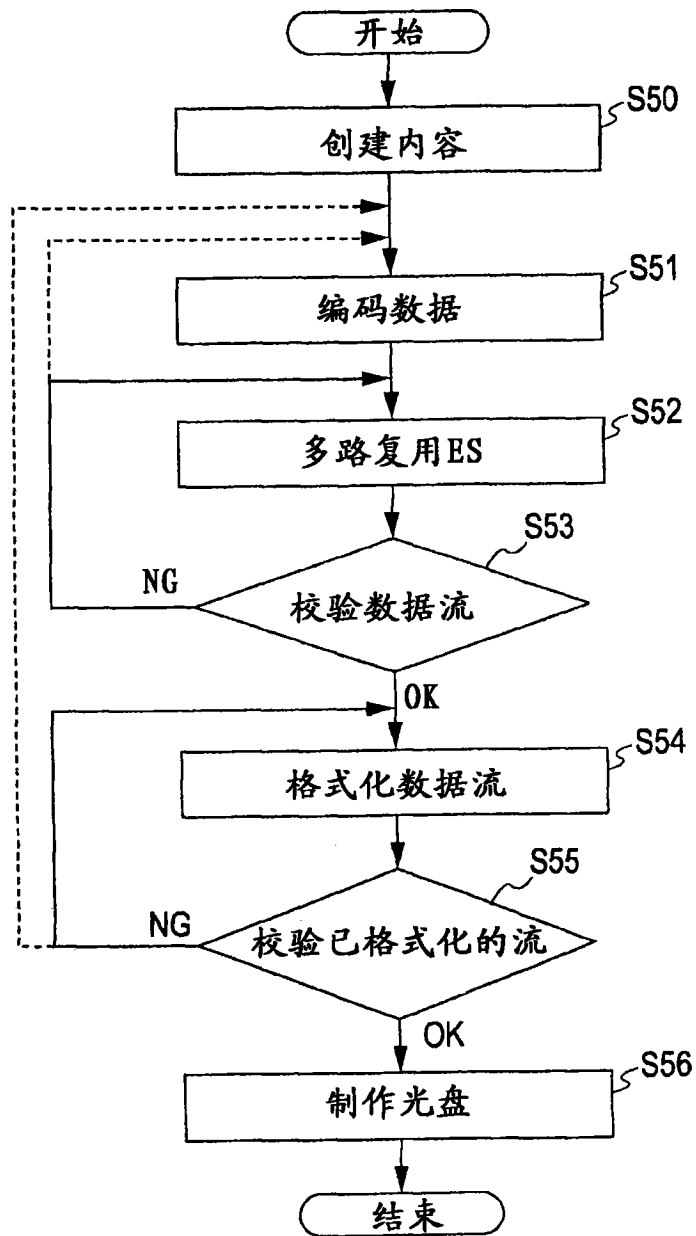


图 38

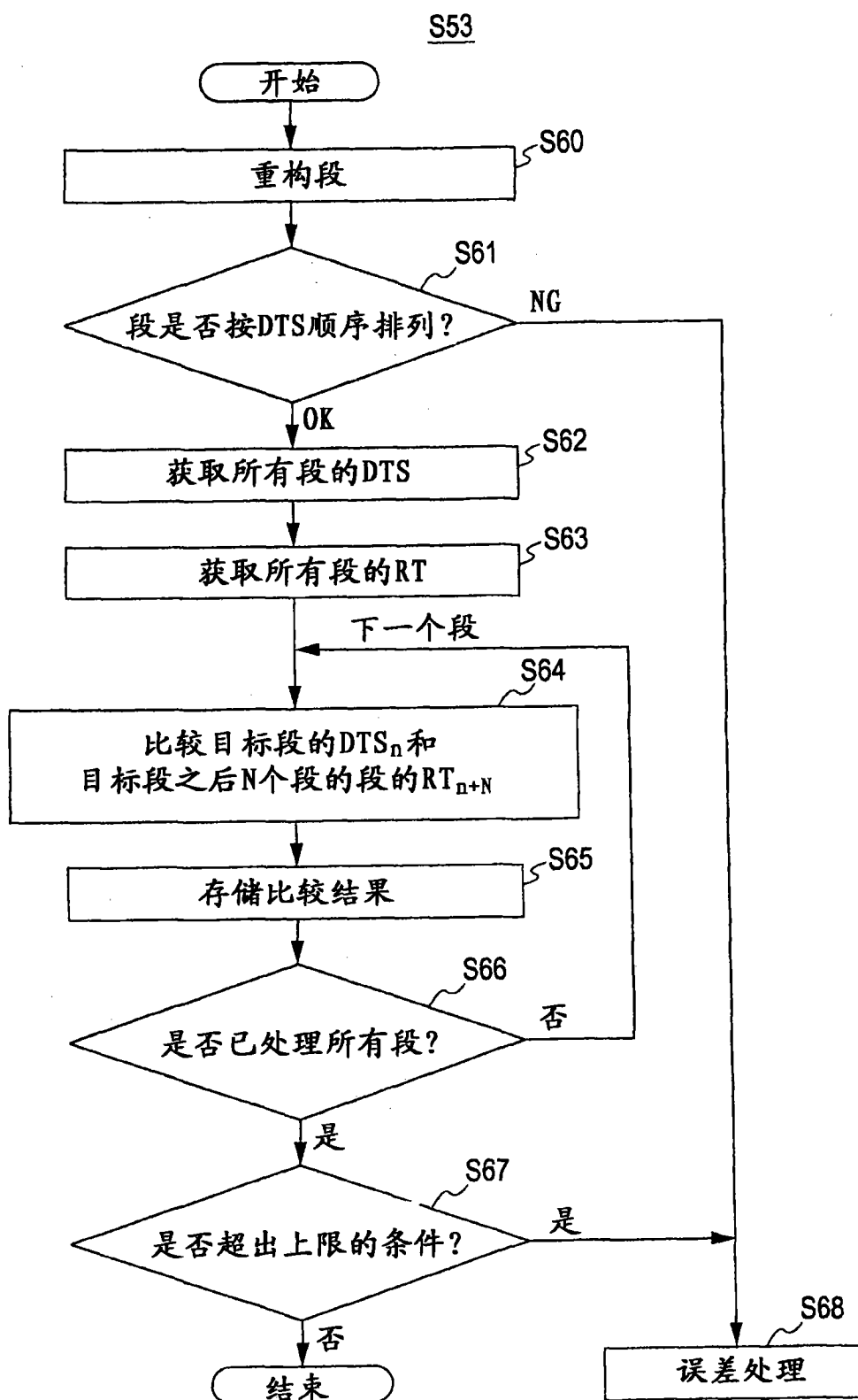


图 39

2

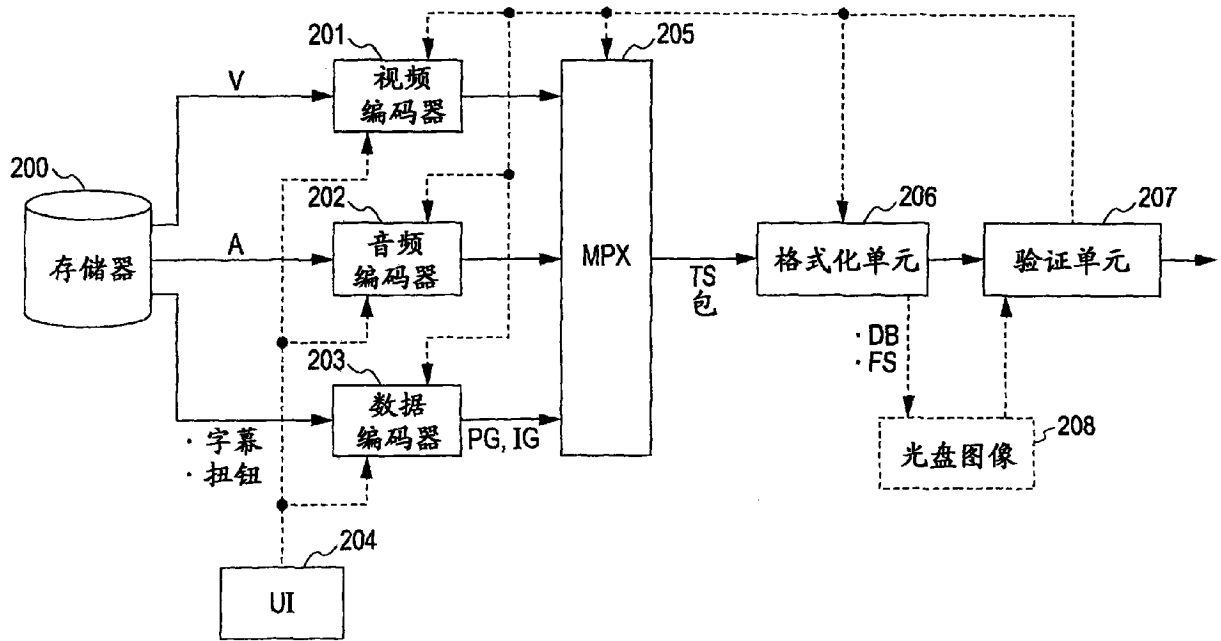


图 40

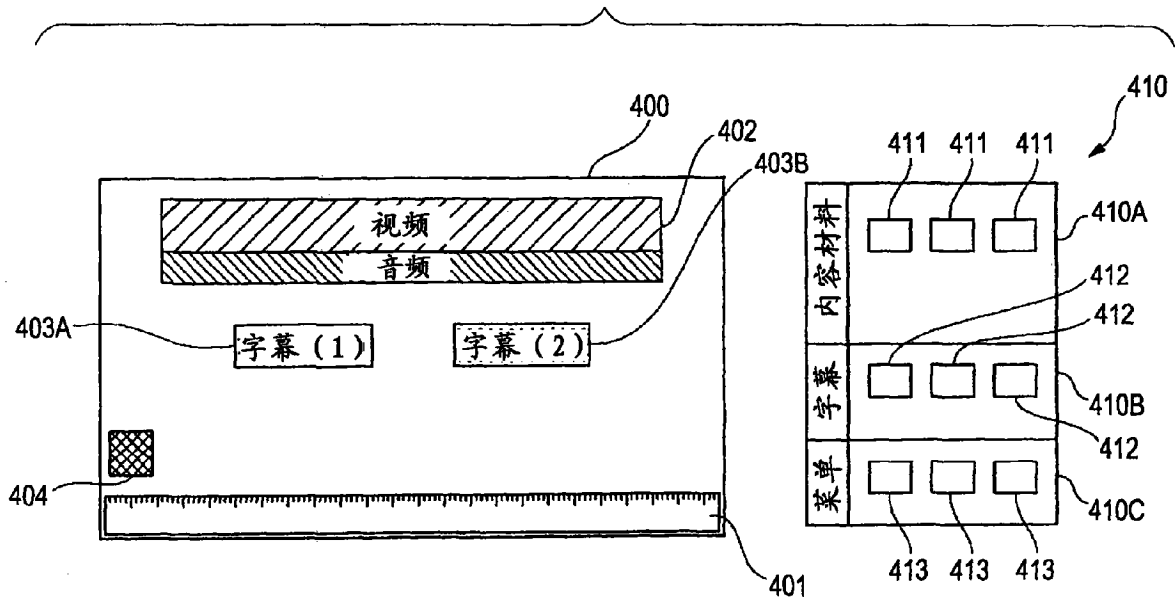


图 41

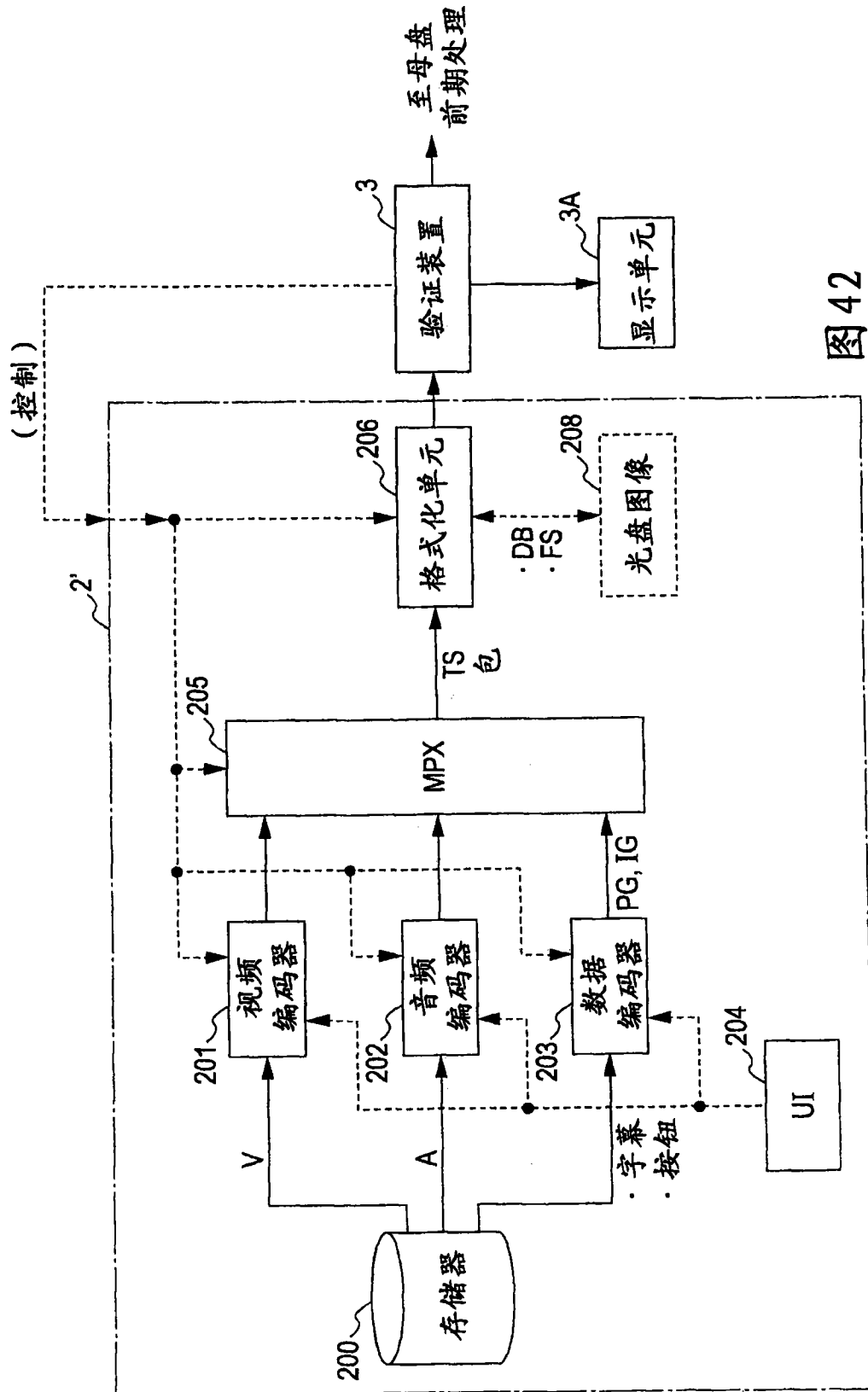


图42

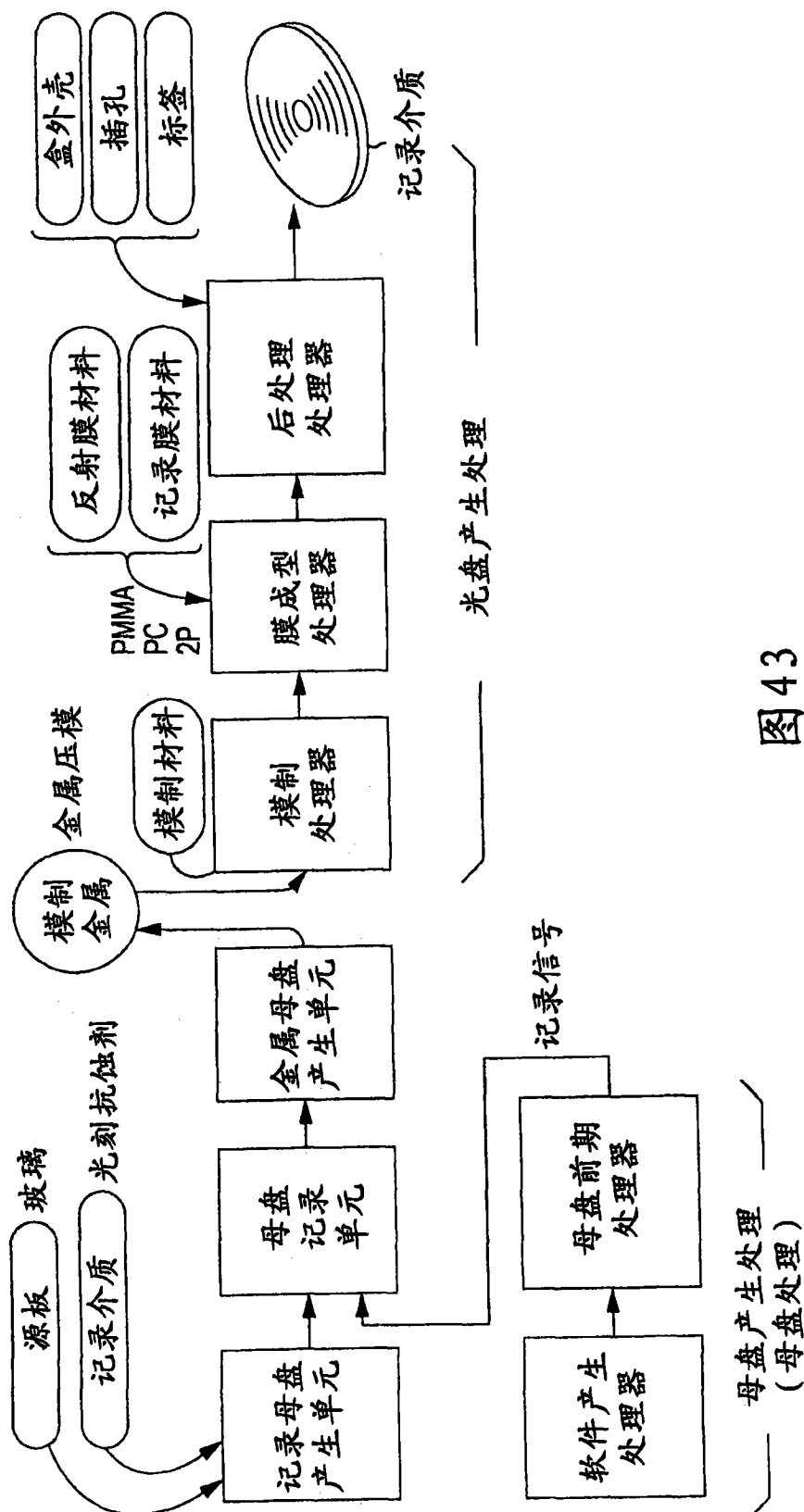


图43