

(12) 发明专利

(10) 授权公告号 CN 101661766 B

(45) 授权公告日 2011.07.27

(21) 申请号 200910172803.4

G11B 20/10(2006.01)

(22) 申请日 2004.11.12

G11B 7/0045(2006.01)

G11B 27/00(2006.01)

(30) 优先权数据

60/519,317 2003.11.12 US

(56) 对比文件

JP 特开平 11-18048 A, 1999.01.22, 全文.

US 2002/0180803 A1, 2002.12.05, 全文.

CN 1144435 A, 1997.03.05, 全文.

(62) 分案原申请数据

200480033446.9 2004.11.12

(73) 专利权人 松下电器产业株式会社

地址 日本大阪府

审查员 马毓昭

(72) 发明人 约瑟夫·麦克罗森 冈田智之

小塚雅之

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 钟胜光 王英

(51) Int. Cl.

H04N 5/92(2006.01)

G11B 7/24(2006.01)

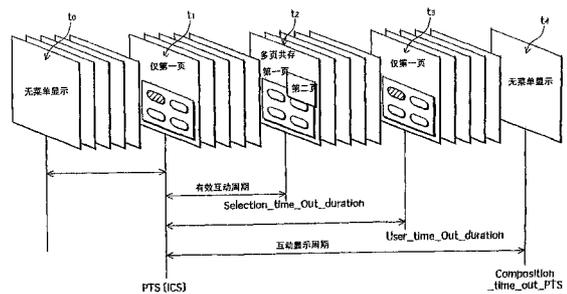
权利要求书 3 页 说明书 35 页 附图 50 页

(54) 发明名称

记录介质、再现装置和方法、记录方法以及计算机可读取程序

(57) 摘要

一种 BD-ROM 存储有视频流和图形流。该视频流代表运动图像。该图形流用于在该运动图像上叠加多页菜单，并且包含互动控制信息 (ICS) 和用于生成该多页菜单的图形数据 (ODS)。附加在包含有互动控制信息的 PES 包中的 PTS 显示了根据视频流再现的进行首次呈现多页菜单的主页的定时。另外，该互动控制信息包含显示超时的信息 (selection_time_out_pts)，其中在发生该超时的情况下自动激活多页菜单的页上的按钮，以及显示了超时的信息 (user_time_out_duration)，其中在发生该超时的情况下自动移除多页菜单的子页。



CN 101661766 B

1. 一种在记录介质上记录图形流和视频流的方法,所述方法包括以下步骤:
生成包括所述视频流和所述图形流的应用数据;以及
将所述应用数据记录在所述记录介质上,其中:
所述生成步骤包括:
第一子步骤,其对视频材料进行编码来获得所述视频流;以及
第二子步骤,其生成一个或多个互动成分部分和多个对象定义部分,从而获得所述图形流,其中每个对象定义部分定义图形对象,
每个互动成分部分是用于定义图形的互动显示成分的功能部分,
在所述第二子步骤中,将超时信息、用户超时信息、用户接口模型信息、和多个页信息描述到每个互动成分部分中,
每个页信息包括一个或多个按钮信息,并且针对多页菜单的相应页定义用于与在所述视频流中包括的画面一起来构成图形的显示成分,
每个按钮信息用于将图形对象呈现为按钮的一个状态,从而将所述多页菜单的每一页呈现为互动显示,
每个按钮都有三种状态,所述三种状态为正常状态、选中状态和激活状态,每个按钮的状态转换顺序为从所述正常状态到所述选中状态、和从所述选中状态到所述激活状态,
在所述第二子步骤中定义的所述超时信息指示在预定的持续时间之后当要激活以所述多页菜单的每个页上的所述选中状态呈现按钮时的时间,
在所述第二子步骤中定义的所述用户接口模型信息指示总是开启模型和弹出模型之一,
如果指示所述总是开启模型,则根据视频再现的进程进行所述多页菜单的状态转换,从而所述多页菜单在显示器上出现或消失,
如果指示所述弹出模型,则响应于用户操作而进行所述多页菜单的状态转换,从而所述多页菜单在显示器上出现或消失,
如果指示所述总是开启模型,则在所述第二子步骤中定义的所述用户超时信息指示将进行到仅第一页显示状态的所述多页菜单的状态转换的时间,以及
如果指示所述弹出模型,则所述用户超时信息指示将进行到无菜单显示状态的所述多页菜单的状态转换的时间。
2. 一种用于再现视频流和图形流的再现装置,所述再现装置包括:
视频解码器,用于对所述视频流进行解码来获得运动图像;以及
图形解码器,用于对所述图形流进行解码来获得图形对象,其中:
所述图形流包括一个或多个互动成分部分和多个对象定义部分,其中每个对象定义部分定义图形对象,
每个互动成分部分是用于定义图形的互动显示成分的功能部分,并包括超时信息、用户超时信息、用户接口模型信息、和多个页信息,
每个页信息包括一个或多个按钮信息,并且针对多页菜单的相应页定义用于与在所述视频流中包括的画面一起来构成图形的显示成分,
每个按钮信息用于将图形对象呈现为按钮的一个状态,从而将所述多页菜单的每一页呈现为互动显示,

每个按钮都有三种状态,所述三种状态为正常状态、选中状态和激活状态,每个按钮的状态转换顺序为从所述正常状态到所述选中状态、和从所述选中状态到所述激活状态,

所述超时信息指示在预定的持续时间之后当要激活以所述多页菜单的每个页上的所述选中状态呈现按钮时的时间,

所述用户接口模型信息指示总是开启模型和弹出模型之一,

如果指示所述总是开启模型,则根据视频再现的进程进行所述多页菜单的状态转换,从而所述多页菜单在显示器上出现或消失,

如果指示所述弹出模型,则响应于用户操作而进行所述多页菜单的状态转换,从而所述多页菜单在显示器上出现或消失,

如果指示所述总是开启模型,则所述用户超时信息指示将进行到仅第一页显示状态的所述多页菜单的状态转换的时间,

如果指示所述弹出模型,则所述用户超时信息指示将进行到无菜单显示状态的所述多页菜单的状态转换的时间,以及

所述图形解码器包括控制器,所述控制器用于基于所述超时信息来执行超时控制,从而

(i) 在由所述超时信息指示的时间,激活在所述多页菜单的当前页上的所述按钮中以所述选中状态呈现的一个按钮,

(ii) 如果指示所述总是开启模型,则在由所述用户超时信息指示的时间,进行到仅第一页显示状态的所述多页菜单的状态转换,以及

(iii) 如果指示所述弹出模型,则在由所述用户超时信息指示的时间,进行到无菜单显示状态的所述多页菜单的状态转换。

3. 一种用于再现视频流和图形流的再现方法,所述方法包括以下步骤:

对所述视频流进行解码来获得运动图像;以及

对所述图形流进行解码来获得图形对象,其中:

所述图形流包括一个或多个互动成分部分和多个对象定义部分,其中每个对象定义部分定义图形对象,

每个互动成分部分是用于定义图形的互动显示成分的功能部分,并包括超时信息、用户超时信息、用户接口模型信息、和多个页信息,

每个页信息包括一个或多个按钮信息,并且针对多页菜单的相应页定义用于与在所述视频流中包括的画面一起来构成图形的显示成分,

每个按钮信息用于将图形对象呈现为按钮的一个状态,从而将所述多页菜单的每一页呈现为互动显示,

每个按钮都有三种状态,所述三种状态为正常状态、选中状态和激活状态,每个按钮的状态转换顺序为从所述正常状态到所述选中状态、和从所述选中状态到所述激活状态,

所述超时信息指示在预定的持续时间之后当要激活以所述多页菜单的每个页上的所述选中状态呈现按钮时的时间,

所述用户接口模型信息指示总是开启模型和弹出模型之一,

如果指示所述总是开启模型,则根据视频再现的进程进行所述多页菜单的状态转换,从而所述多页菜单在显示器上出现或消失,

如果指示所述弹出模型,则响应于用户操作而进行所述多页菜单的状态转换,从而所述多页菜单在显示器上出现或消失,

如果指示所述总是开启模型,则所述用户超时信息指示将进行到仅第一页显示状态的所述多页菜单的状态转换的时间,

如果指示所述弹出模型,则所述用户超时信息指示将进行到无菜单显示状态的所述多页菜单的状态转换的时间,以及

在对所述图形流进行解码的步骤中,基于所述超时信息来执行超时控制,从而

(i) 在由所述超时信息指示的时间,激活在所述多页菜单的当前页上的所述按钮中以所述选中状态呈现的一个按钮,

(ii) 如果指示所述总是开启模型,则在由所述用户超时信息指示的时间,进行到仅第一页显示状态的所述多页菜单的状态转换,以及

(iii) 如果指示所述弹出模型,则在由所述用户超时信息指示的时间,进行到无菜单显示状态的所述多页菜单的状态转换。

记录介质、再现装置和方法、记录方法以及计算机可读取程序

[0001] 本申请是申请日为 2004 年 11 月 12 日、申请号为“200480033446.9”并且发明名称为“记录介质、再现装置和方法、记录方法以及计算机可读取程序”的在中国递交的专利申请的分案申请。

发明领域

[0002] 本发明涉及用户界面控制的领域。更具体地,本发明涉及用于消费者电影发行的记录介质的改进,以及具有一种用户界面并且用于再现记录介质的消费者再现装置的改进。

技术背景

[0003] 使用菜单结构的用户界面是用于接收用户操作的一种必要的功能,其中这些用户操作例如选择再现的语言和章节。

[0004] 如果一部电影提供了多种可由用户选择的选项,例如多个章节,那么可能会出现无法将全部选项都呈现在单个屏幕显示器上的情况。因此,就需要一种多页菜单,其中通过该菜单可以响应于用户操作按照顺序呈现多个页,从而可以在菜单的各个页上适当地呈现多个选项。

[0005] 但是,不幸地是,如果显示了太多页,那么这些页就会占据屏幕的一大部分,由此会影响到用户观看同时也被显示在屏幕上的主电影。如果用户重复地作出操作以移除他不再感兴趣的每页,那么就可以避免上述问题的发生。但是,如果需要用户反复地进行上述这种令人生烦的操作,那么这又会成为另一种打扰用户观看电影的因素。

[0006] 另外,DVD 所提供的互动控制具有下述劣势。对于将要被显示的每个菜单页来说,需要对 DVD 进行访问以便读取页信息。由于这些访问,无法在不打断视频再现的情况下显示菜单页。而这些打断会使得用户难以专心致志地欣赏电影。

发明内容

[0007] 本发明的目的在于提供一种其上记录有视频流和图形流的记录介质,在该记录介质中:所述图形流包括一个或多个互动成分部分和对象定义部分,每个对象定义部分定义图形对象;每个互动成分部分包括多个页信息;所述多个页信息定义多页菜单的多个页的显示成分,并且每个页信息包括一个或多个按钮信息;每个按钮信息用于将图形对象呈现为按钮材料的一个状态,从而将所述多页菜单的每一页呈现为互动显示;每个互动成分部分被封装在附加有呈现时间标记的包中;以及所述呈现时间标记在所述视频流的再现时间轴上显示时间,在所述时间,所述多页菜单的第一页变为可用于弹出显示,从而所述多页菜单的状态从无菜单显示状态改变到仅第一页显示状态。

[0008] 附图简述

[0009] 图 1 显示了根据本发明的记录介质的使用方式;

- [0010] 图 2 显示了 BD-ROM 的一种示例性结构；
- [0011] 图 3 示意性地显示了一种 AV 剪辑的结构；
- [0012] 图 4A 显示了互动图形流的结构；
- [0013] 图 4B 显示了载有多个功能部分的 PES 包；
- [0014] 图 5 显示了由各种类型的功能部分所规定的逻辑结构；
- [0015] 图 6 显示了被分配了 DS_n 的 AV 剪辑再现的时间轴；
- [0016] 图 7A 和 7B 显示了 interactive_composition 结构和 ICS 之间的关系；
- [0017] 图 8 显示了 ICS 的内部结构；
- [0018] 图 9 显示了在 stream_model 域显示为“复用”类型的情况下对 ICS 执行的处理；
- [0019] 图 10 显示了在 stream_model 域显示为“预先加载”类型的情况下对 ICS 执行的处理；
- [0020] 图 11 是一个时序图,显示了在 DS_n 是时元 (Epoch) 开始 $DS(n = 1)$ 的情况下,属于该 DS_n 的 ICS 中的 selection_time_out_pts 和 composition_time_out_pts 字段的功能；
- [0021] 图 12A 显示了在使用弹出 U/I 情况下多页菜单的状态转换；
- [0022] 图 12B 显示了在使用总是开启 U/I 情况下多页菜单的状态转换；
- [0023] 图 13 是包含多页菜单的显示情况；
- [0024] 图 14A-14C 显示了由 selection_time_out_pts 规定的显示例子；
- [0025] 图 15A-15D 显示了由 user_time_out_duration 规定的显示例子；
- [0026] 图 16 显示了相关于多页菜单的多个页中任意一页 (第 x 页) 的页信息的内部结构；
- [0027] 图 17 显示了按钮 0-A、0-B、0-C 和 0-D 的状态转换；
- [0028] 图 18 显示了规定了图 17 中所示按钮状态转换的 button_info 描述的例子；
- [0029] 图 19 显示了通过 ODS 11-19 得到的图形图像；
- [0030] 图 20 显示了一种 effect_sequence 结构；
- [0031] 图 21A 示意性地显示了如何确定 effect_duration；
- [0032] 图 21B 显示了计算 effect_duration 的等式；
- [0033] 图 22 显示了一种引入效果动画的具体例子；
- [0034] 图 23 显示了根据 window_width、window_height 和 effect_duration 域值所呈现的退出效果动画；
- [0035] 图 24 详细显示了任意一个 composition_object(i) 的内部结构；
- [0036] 图 25 显示了一种代表右滚动动画效果的引入效果的具体例子；
- [0037] 图 26 显示了实现图 25 中所示引入效果的 compositon_object(0) 结构的例子；
- [0038] 图 27 显示了实现图 25 中所示引入效果的 compositon_object(1) 结构的例子；
- [0039] 图 28 显示了实现图 25 中所示引入效果的 compositon_object(2) 结构的例子；
- [0040] 图 29 显示了实现图 25 中所示引入效果的 compositon_object(3) 结构的例子；
- [0041] 图 30 显示了 DS_n 内的 ODS 的顺序；
- [0042] 图 31 显示了在 default_selected_button_id_ref 域被设置为“00”和被设置为指定了按钮 B 的一个有效值这些不同情况下,S-ODS 内的 ODS 之间的顺序的差别；
- [0043] 图 32A 显示了在静态地确定了缺省选择按钮的情况下的一种 LASTODSPTS；

- [0044] 图 32B 显示了在动态地确定了缺省选择按钮的情况下的一种 LASTODSPTS ;
- [0045] 图 33A 显示了在 DS 中不包含与选中状态相关的 ODS 情况下的一种 LASTODSPTS ;
- [0046] 图 33B 显示了在 DS 中并不包含与正常状态相关的 ODS 情况下的一种 LASTODSPTS ;
- [0047] 图 33C 显示了在 DS 中根本不包含 ODS 情况下的一种 LASTODSPTS ;
- [0048] 图 34A 显示了用于获得 DECODEDURATION 值的算法,其中在该算法中利用了通过 $LASTODSPTS(DS_n) - DTS(DS_n[ICS])$ 计算得到值和 PLANECLERATIME 的值 ;
- [0049] 图 34B 和 34C 显示了如何计算 DECODEDURATION 值的例子 ;
- [0050] 图 35A 显示了用于计算 EFFECTTD 的等式 ;
- [0051] 图 35B 显示了在未提供引入效果并且静态地确定了缺省选择按钮的情况下用于计算 PAGDEFAULTTD 的等式 ;
- [0052] 图 35C 显示了在未提供引入效果并且动态地确定了缺省选择按钮的情况下用于计算 PAGENODEFAULTTD 的等式 ;
- [0053] 图 36 显示了用于有选择地确定 EFFECTTD、PAGEDEFAULTTD 和 PAGENODEFAULTTD 中的一个作为 TRANSFERDURATION 的一种算法 ;
- [0054] 图 37 显示了根据本发明的再现装置的内部结构 ;
- [0055] 图 38A 是对 PSR 11 所执行的“再现条件改变时执行的过程”的流程图 ;
- [0056] 图 38B 是对 PSR 11 所执行的“要求改变时执行的过程”的流程图 ;
- [0057] 图 39A 是对 PSR 10 所执行的“再现条件改变时执行的过程”的流程图 ;
- [0058] 图 39B 是对 PSR 10 所执行的“要求改变时执行的过程”的流程图 ;
- [0059] 图 40 显示了 I 图形解码器 9 的内部结构 ;
- [0060] 图 41 是图形控制器 37 所执行的过程的主程序的流程图 ;
- [0061] 图 42 是用于首次呈现的过程的流程图 ;
- [0062] 图 43 是用于再现引入效果动画的过程的流程图 ;
- [0063] 图 44 是用于按照动画的方式呈现按钮的显示更新过程的流程图 ;
- [0064] 图 45 是用于执行导航命令的过程的流程图 ;
- [0065] 图 46 显示了用于处理 UO 的过程 ;
- [0066] 图 47 是当前按钮改变过程的流程图 ;
- [0067] 图 48 是用于按钮的自动激活的过程的流程图 ;以及
- [0068] 图 49 是根据第一到第五实施例的 BD-ROM 的制造过程的流程图。

[0069] 发明详述

[0070] (第一实施例)

[0071] 接下来,将描述根据本发明的记录介质的一种实施例。首先,将描述如何使用该记录介质,而这是实施本发明的一种形式。图 1 显示了根据本发明的记录介质的使用方式。在该图中,BD-ROM 100 是根据本发明的一种记录介质。BD-ROM 100 用于向包含再现装置 200、遥控器 300 和电视 400 的家庭影院系统提供电影。

[0072] 上面就是对根据本发明的介质的使用方式的描述。接下来,将描述如何生产该记录介质,而这是实施本发明的另一种方式。可以通过改进 BD-ROM 的应用层从而实现该记录介质。图 2 显示了 BD-ROM100 的一种示例性结构。

[0073] 在该图中,BD-ROM 100 位于第四级,而 BD-ROM 的轨道位于第三级。尽管在该图中

将轨道展开为一条直线,但是在实际应用中轨道是从BD-ROM的中心螺旋向外。轨道包含导入区、内容区和导出区。内容区具有包含物理层、文件系统层和应用层的层模型。第一级按照目录结构的形式显示了BD-ROM的应用层的格式(应用格式)。

[0074] 第一级显示了应用层的格式(应用格式)。在该图中,BD-ROM具有一个ROOT目录、位于ROOT目录下的BDWV目录以及位于BDMV目录下的STREAM目录。

[0075] STREAM目录包含一个AV剪辑和多个子剪辑。AV剪辑和子剪辑位于STREAM目录的文件中,例如“00001.m2ts”、“00002.m2ts”和“00003.m2ts”。STREAM目录中的每个文件都具有统一格式的文件名,即“xxxxx.m2ts”,其中x是任意整数。

[0076] <AV剪辑结构>

[0077] 接下来将描述AV剪辑。图3示意性地显示了AV剪辑的结构。

[0078] 在该图中,中间级显示了AV剪辑。可以按照下述方式生成该AV剪辑。上第一级所显示的包含多个视频帧(画面pj1、pj2、pj3,...)的视频流和包含多个音频帧的音频流被分别转换为上第二级所示的PES包,然后进一步被转换为上第三级所示的TS包。同样地,下第一级所示的呈现图形流(此后称作“PG流”)和互动图形流(此后称作“IG流”)被分别转换为下第二级所示的PES包,然后进一步被转换为下第三级所示的TS包。音频流和视频流、PG流以及IG流的这些TS包被复用到一起从而形成AV剪辑。

[0079] PG流和IG流是流格式的数据,用于与视频流的每个画面一起构成图形。PG流和IG流所载有的用于叠加到画面上的数据实体被称作“图形对象”。

[0080] PG流是一种用于通过图形对象呈现字幕的图形流。

[0081] IG流是一种用于互动地呈现图形对象的图形流。上述就是对AV剪辑结构的描述。接下来将描述AV剪辑和子剪辑。子剪辑是未经复用的。子剪辑是仅包含IG流、音频流和文本数据的文件。被存储为子剪辑的IG流被预先加载到存储器中,以便实现与AV剪辑的同步再现。这种预先将子剪辑加载到存储器中的操作被称为“预先加载”。

[0082] 接下来,将描述IG流。图4A显示了IG流的结构。在第一级,显示了AV剪辑所包含的一列TS包。在第二级,显示了图形流所包含的一列PES包。通过将第一级中所示TS包集合内具有预先确定的PID的TS包的载荷连接到一起从而形成第二级中所示的PES包。由于PG流并不是本发明的重点,所以我们不对PG流进行深入的描述。

[0083] 在第三级,显示了图形流的结构。图形流包含这样一些功能部分,其中这些功能部分包含ICS(互动成分部分)、PDS(调色板定义部分)、ODS(对象定义部分)以及END(显示集合结束部分)。在这些功能部分中,ICS是一种显示成分部分,而PDS、ODS和END是定义部分。每个功能部分与PES包之间的关系是一对一或是一对多的。也就是说,将功能部分转换为单个PES包,或者被分为几部分然后被转换为多个PES包之后,将一个功能部分记录到BD-ROM 100上。

[0084] 接下来将逐一描述这些功能部分。

[0085] ICS是用于规定图形的互动显示成分的功能部分。互动显示部分的一个例子是多页菜单。

[0086] ODS是一种包含多个行程长度数据的行程长度编码图形对象。行程长度数据使用像素编码表示了一个像素串,其中该像素编码显示了像素值和该像素值的连续长度。像素编码是一种显示了从1到255中一个值的8位值。通过使用这种像素编码,行程长度数据

可以设置全色彩 (16777216 种色彩) 中的任意 256 种像素色彩。

[0087] PDS 是用于存储调色板数据的功能部分。调色板数据显示了从 1 到 255 的像素编码和像素值的组合。这里所使用的像素值包含红色差别成分 (Cr 值)、蓝色差别成分 (Cb 值)、亮度成分 (Y 值) 以及透明度 (T 值)。通过将每个行程长度数据的像素编码替换到调色板数据所显示的像素值中会产生一种色彩。

[0088] END 提供了一种明确的指示,它指示已经完成了功能部分的传输。END 被作为最后的部分插入到图形流中。上述是对每种功能部分的描述。

[0089] 图 4B 显示了载有功能部分的 PES 包。如图中所示,每个 PES 包都包含一个包头和有效载荷。有效载荷是载有功能部分的实体,并且包头载有与该功能部分相关的一个 DTS 和 PTS。此后,将载有功能部分的 PES 包的包头中的 DTS 和 PTS 简单地称作该功能部分的 DTS 和 PTS。

[0090] 这些多种类型的功能部分规定了一种例如图 5 中所示的逻辑结构。在该图中,第一级显示了多个时元,第二级显示了 DS (显示集合),而第三级显示了 DS 的类型。在该图中的第四级上显示了图 4A 中第三级上所示的功能部分。

[0091] 第一级上的时元是指 AV 剪辑再现的时间轴上连续存储器管理的一个时间单位,并且也是指分配给该时间单位的一组数据。这里所说的存储器包含用于存储一个屏幕的图形对象的图形平面和用于存储未压缩图形数据的对象缓存器。连续存储器管理意味着在整个时元内图形平面和对象缓存器都不会被清理,并且仅在图形平面上一个预定矩形区域内执行图形的清除和绘制 (“清理”意味着清除整个图形平面或整个对象缓存器)。在整个时元内该矩形区域的尺寸和位置都是固定的。只要是在图形平面上该固定的矩形区域内执行清除和绘制图形,那么就可以保证实现无缝再现。也就是说,该时元是 AV 剪辑再现的时间轴上无缝再现的时间单位。为了改变图形平面上的图形清除 / 绘制区域,需要在再现时间轴上规定一个点,并且在该点处进行改变以及从该点开始向前设置一个新的时元。在这种情况下,两个时元之间的边界就不是无缝的。

[0092] 这里所提到的无缝再现是指在预先确定数量的视频帧内完成图形的清除和绘制。对于 IG 流,视频帧的数量是四到五个。根据固定矩形区域与整个图形平面之间的比例和对象缓存器和图形平面之间的传送速率从而确定视频帧的数量。

[0093] 第二级上的 DS 是包含在图形流中的一组功能部分,并且提供了一个完整的显示成分。在图 5 中,虚线 hk1 显示了第二级上的 DS 是属于哪个时元。DS₁、DS₂、DS₃... ,和 DS_n 属于第一级上的第一个时元。

[0094] 第三级显示了 DS 的类型。时元中的第一个 DS 是一种“时元开始”DS。该时元中第一个 DS 之外的其它每个 DS 要么是一种“获取点”DS,要么是“正常情况”DS 或者“时元继续”DS。尽管按照“获取点”、“正常情况”和“时元继续”这种顺序显示了这些 DS,但是该顺序仅是一种例子,并且可以按照其它不同的顺序排列这些 DS。

[0095] 时元开始 DS 提供了一种显示效果“新显示”,并且指示了一个新时元的开始。因此,时元开始 DS 包含用于下一个屏幕成分的所有必要的功能部分。时元开始 DS 位于这样一个点:有可能在该点进行跳读操作,例如电影中一个章节的开始。

[0096] 获取点 DS 提供了一种显示效应“显示刷新”,并且在内容上与先前的时元开始 DS 相同。获取点 DS 并不是时元的开始,但是却包含了用于下一个屏幕成分的所有必要的功能

部分。因此,当再现是从获取点 DS 开始进行时,可以可靠地显示图形。也就是说,获取点 DS 使得可以从时元中的中间点得到一个显示成分。

[0097] 获取点 DS 位于这样一个点:可能在该点进行跳读操作,例如可以由时间搜索指定的点。该时间搜索是一种对应于用户按照分钟/秒的方式所输入的时间从而定位再现点的操作。可以按照相对较大的单位进行时间输入,例如十分钟和十秒钟。因此,可以在这样一个时间点提供获取点 DS:可以通过以 10 分钟和 10 秒钟为单位所进行的时间搜索从而指定该时间点。通过在可以由时间搜索指定的这样一种时间点提供获取点 DS,当进行时间搜索时可以平滑地执行图形流的再现。

[0098] 正常情况 DS 提供了一种显示效果“显示更新”,并且仅包含与前面显示成分的不同之处。例如,如果 DS_v 在内容上与前一个 DS_u 相同,但是在显示成分上有差别,那么 DS_v 就是仅包含一个 ICS 或 ODS 的正常情况 DS。这就使得不需要在 DS 中提供重叠的 ODS,从而可以降低存储在 BD-ROM 100 中的数据量。因为正常情况 DS 仅包含差别而不包含重叠数据,所以仅通过正常情况 DS 并不能显示任何显示成分。

[0099] 时元继续 DS 指示了在一个 AV 剪辑的开始处,一个时元是从另一个 AV 剪辑继续而来。例如, DS_n 包含一个成分状态字段,该字段的值指示了“时元继续”。在这种情况下,即使该 DS_n 与上一个 DS_{n-1} 位于不同的 AV 剪辑上,这两个 DS_n 也同属于一个时元。因为 DS_n 和 DS_{n-1} 属于同一个时元,所以即使在这两个 DS_n 之间存在 AV 剪辑的边界,也不会清理图形平面和对象缓存器。

[0100] 在图 5 中,虚线 kz1 显示了第四级上的功能部分属于哪一个 DS。因为图 4A 和 5 中所示的功能部分是相同的,所以图 4A 中所示的功能部分属于时元开始 DS。属于获取点 DS 的功能部分与属于时元开始 DS 的功能部分相同。另外,属于正常情况 DS 的功能部分与属于时元开始 DS 的功能部分相同,不同之处只是在正常情况 DS 中省略了一些功能部分。

[0101] 上面就是对功能部分所规定的逻辑结构的描述。接下来描述将具有这样的 ICS 和多个 ODS 的 DS 分配到 AV 剪辑再现的时间轴上。因为时元是一种连续存储器管理的时间单位,并且包含一个或多个 DS,所以这些 DS 是如何分配到 AV 剪辑再现的时间轴上就是很重要的。注意到“AV 剪辑再现的时间轴”是指用于规定被复用到 AV 剪辑中的单个画面的解码定时和再现定时的时间轴。解码定时和再现定时的时间精度为 90KHz。DS 中的 ICS 或 ODS 的 DTS 和 PTS 显示了实现同步控制所基于的定时。根据被附加到 ICS 和 ODS 的每个 DTS 和 PTS 进行同步控制,由此将 DS 分配到 AV 剪辑再现的时间轴上。

[0102] 假设用 DS_n 表示一个时元内的任意一个 DS,并且根据图 6 中所示的 DTS 和 PTS 集合将该 DS_n 分配到 AV 剪辑再现的时间轴。

[0103] 图 6 显示被分配了 DS_n 的 AV 剪辑再现的时间轴。在该图中, DS_n 的持续时间在属于 DS_n 的 ICS 的 DTS 值 ($DTS(DS_n[ICS])$) 所指示的时间处开始,在属于 DS_n 的 END 的 PTS 值 ($PTS(DS_n[ICS])$) 所指示的时间处结束。ICS 的 PTS 值 ($PTS(DS_n[ICS])$) 显示了 DS_n 的首次呈现的显示时间。通过调整 $PTS(DS_n[ICS])$ 值,使其对应于视频流中一个希望得到的画面的显示定时, DS_n 的首次呈现就可以与该视频流同步。

[0104] $PTS(DS_n[ICS])$ 值是对 ODS 进行解码所花费的时间 (DECODEDURATION) 和转移解码后的图形对象 (TRANSFERDURATION) 所花费的时间之和。

[0105] 在 DECODEDURATION 内完成对首次呈现所需的 ODS 的解码。图 6 所示的持续时间

mc1 是对 ODS_m 进行解码所需的时间,其中 ODS_m 是 DS_n 内任意一个 ODS。 $DTS(ODS_n[ODS_m])$ 的值显示了解码持续时间 mc1 的开始,并且 $PTS(ODS_n[ODS_m])$ 的值显示了解码持续时间 mc1 的结束。

[0106] 如上面所述,通过将时元内的每个 ODS 分配到 AV 再现的时间轴从而规定该时元。上面就是对分配到 AV 再现的时间轴的描述。

[0107] 第一实施例的一个特征在于根据 AV 再现的进程从而控制多页菜单的动作。用于实现该特征的新颖结构是通过 ICS 中提供的 `interactive_composition` 结构实现的。接下来将描述 `interactive_composition` 结构。

[0108] 图 7A 和 7B 显示了 `interactive_composition` 结构和 ICS 之间的关系。`interactive_composition` 结构和 ICS 可以是图 7A 中所示的一一对应的关系,或者是图 7B 中所示的一对多的对应关系。

[0109] 一一对应适用于 `interactive_composition` 结构足够小从而可以被包含在一个 ICS 中的情况。

[0110] 一对多对应则适用于这样一种情况:其中 `interactive_composition` 结构的尺寸太大,以致无法由一个 ICS 携带,而只能被分为几段从而存储在多个 ICS 中。因为 `interactive_composition` 结构可以被分为几段从而存储在多个 ICS 中,所以对 `interactive_composition` 结构的大小尺寸就没有限制。`interactive_composition` 结构可以为 512K 字节、1M 字节或任意数量的字节。如上面所述,一个 `interactive_composition` 结构可以对应于多个 ICS。但是,为了简便起见,此后所进行的描述都是基于这样一种前提:ICS 和 `interactive_composition` 结构是一一对应的对应关系。

[0111] 图 8 显示了 ICS 的内部结构。ICS 包含一个 `interactive_composition` 结构或一个 `interactive_composition` 结构的一部分。如图 8 的左侧部分所示,ICS 包含下述字段:显示 ICS 的部分类型的“`segment_descriptor`”;显示用像素表示的视频平面的宽度和高度以及相关视频流的帧速率的“`video_descriptor`”;显示 `composition_state` 的“`composition_descriptor`”,它是显示了 ICS 所属于的 DS 的类型是正常情况 DS、获取点 DS 和时元开始 DS 或者 `Effect_Sequence` 中的哪一种的信息;以及显示 ICS 究竟是载有整个 `interactive_composition` 结构还是只载有 `interactive_composition` 结构的一部分的“`interactive_compositon_data_fragment`”。

[0112] 如箭头 cu1 所示,图 8 详细显示了 `interactive_composition` 结构的内部结构。`interactive_composition` 结构包含下述字段:“`interactive_compositon_length`”、“`stream_model`”、“`user_interface_model`”、“`composition_time_out_pts`”、“`selection_time_out_pts`”、“`user_time_out_duration`”,以及用于可供显示的多页菜单的每一页的“`page_information(1), (2)... (i)... (number_of_page-1)`”。

[0113] “`interactive_compositon_length`”字段显示了 `interactive_composition` 结构的长度。

[0114] “`stream_model`”字段显示了相关的 `interactive_composition` 结构的流模型的类型。流模型显示了如何将 `interactive_composition` 结构存储到 BD-ROM 中以及再现装置中的缓存器(称作“成分缓存器”)如何对数据进行处理。具体而言,流模型显示了(i)图形流究竟是与 AV 剪辑复用到一起并且将其分离加载到成分缓存器中,还是(ii)将图形流

作为子剪辑与 AV 剪辑存储在不同的文件中并且将其预先加载到成分缓存器中。之所以在 `interactive_composition` 结构中配备“`stream_model`”字段是因为成分缓存器需要根据 `interactive_composition` 结构是被预先加载的还是被复用的从而执行不同的处理。图 9 显示了对“复用”ICS 所进行的处理。在该图中,在第四级上显示了被复用到 AV 剪辑中的视频流,并且在第三级上显示了被复用到 AV 剪辑中的图形流。该图形流包含三个 DS,即 DS_1 、 DS_2 和 DS_8 。第二级显示了 AV 再现的进行。第一级显示了在 AV 再现的不同时间处存储在成分缓存器中的内容。如果当前再现点到达被复用到 AV 剪辑中的 DS_1 的开始处,那么 DS_1 被加载到成分缓存器,如箭头 pf1 所示。因此,成分缓存器存储 DS_1 [ICS]。然后,如果当前再现点到达被复用到 AV 剪辑中的 DS_2 的开始处,那么 DS_2 被加载到成分缓存器,如箭头 pf2 所示。因此,成分缓存器存储 DS_2 [ICS]。也就是说,如果 `stream_model` 是“复用”,那么就会存在多于一个的 `interactive_composition` 结构。再现装置丢弃当前所存储的 `interactive_composition` 结构,并且将位于当前再现点处的另一个 `interactive_composition` 结构最新地加载进来。因此,成分缓存器一次只能存储一个 `interactive_composition` 结构。

[0115] 当进行跳读操作时,这种丢弃和加载操作特别有效。当进行跳读操作时,就不再需要与当前再现点相关的 `interactive_composition` 结构。因此,再现装置丢弃当前位于成分缓存器中的 `interactive_composition` 结构。如图 9 中所示,当执行跳读操作 am1 时,成分缓存器丢弃在跳读操作的执行时间 am2 时存储的内容。然后,位于跳读目的地处的 DS_8 [ICS] 被加载到成分缓存器中,如箭头 pf3 所示。

[0116] 如上面所述,在“复用”类型流模型的情况下,随着 AV 再现的进行,成分缓存器需要丢弃所存储的 `interactive_composition` 结构。与此相反,在“预先加载”类型流模型的情况下,成分缓存器则不再需要丢弃所存储的 `interactive_composition` 结构。如果随着 AV 再现的进行,成分缓存器丢弃了存储的 `interactive_composition` 结构,那么需要再次从 BD-ROM 加载与被丢弃的相同的 `interactive_composition` 结构。这种无用的重新加载相同的 `interactive_composition` 结构的操作会导致额外的负担。图 10 显示了成分缓存器的内容。图 10 和 9 中的第一级到第四级上所示的内容是相同的。但是在图 10 和 9 中,不同之处在于在不同的时间处,成分缓存器所存储的内容完全不同。具体而言,即使当前再现点到达 DS_1 被复用的地方, DS_1 的 ICS 也不会被加载到成分缓存器中(由图中标记“X”指示)。

[0117] 类似地,即使当前再现点到达 DS_2 被复用的地方, DS_2 的 ICS 也不会被加载到成分缓存器中(由图中标记“X”指示)。类似地,即使执行了跳读操作,存储在成分缓存器中的内容也会被保留而不是被丢弃。`interactive_composition` 结构具有 `stream_model` 字段,用于实现控制对象(图形控制器),以便准确地判断随着 AV 再现的进行究竟是丢弃所存储的 `interactive_composition` 结构,还是无论 AV 再现的进行情况如何都保留所存储的 `interactive_composition` 结构。上述是对 `stream_model` 字段的描述。

[0118] 再次参考图 8,“`user_interface_model`”显示了用于 `interactive_composition` 结构的用户界面模型的类型。`user_interface_model` 显示了“总是开启 U/I”或“弹出 U/I”。如果使用的是总是开启 U/I,那么随着 AV 剪辑再现的进行,菜单出现在屏幕上并且消失。如果使用弹出 U/I,那么菜单响应于用户操作出现在屏幕上并且消失。

[0119] `composition_time_out_pts` 字段显示了 ICS 所属的时元的结束时间(时元结

束)。在时元结束时,由 ICS 所规定的互动控制不再有效。也就是说,由 `compostion_time_out_pts` 字段所显示的时间点是有效互动控制周期的结束。

[0120] “`selection_time_out_pts`” 字段显示了按钮选择终止时间。在由 `selection_time_out_pts` 值所示的时间,自动激活当前被选中的按钮。注意到按钮是代表多页菜单中的选项的元素。`selection_time_out_pts` 字段规定了用户通过互动从而激活按钮的有效互动周期。

[0121] 图中显示的 IF 表述 (`if(stream_model == “0b”)`) 意味着上述的“`compostion_time_out_pts`”和“`selection_time_out_pts`”字段是可选信息,并且仅当“`stream_model`”字段显示为“复用”类型时才提供它们。如果 ICS 的 `stream_model` 为“预先加载”类型,那么就不会提供“`compostion_time_out_pts`”和“`selection_time_out_pts`”字段。

[0122] `user_time_out_duration` 字段显示了超时,其中如果到达该超时时间,就移除响应于用户操作从而呈现到屏幕上的页面。如果使用“总是开启 U/I”,那么响应于用户的操作,呈现第二页及其之后 (`onward`) 的页(称作子页)。因此,在由 `user_time_out_duration` 所规定的超时时间之后,就从屏幕上移除所有子页,而只在屏幕上保留第一页。如果使用“弹出 U/I”,那么就响应于用户操作,不仅可以将子页而且可以将多页菜单上的每一页呈现在屏幕上。因此,在 `user_time_out_duration` 之后,所有屏幕上的页都会被移除,而屏幕上不保留任意一页(无菜单显示)。

[0123] 接下来将描述时元中 `selection_time_out_pts` 和 `compostion_time_out_pts` 字段的含义。

[0124] 图 11 是一个时序图,显示了其中在 DS_n 是时元开始 $DS(n = 1)$ 的情况下,属于 DS_n 的 ICS 中的 `selection_time_out_pts` 和 `compostion_time_out_pts` 字段的功能。该时序图显示了由 `compostion_time_out_pts` 值指定时元的结束(时元结束)。互动显示对于从 $PTS(DS_n[ICS])$ 到该时元结束的周期内是有效的。该周期被称作“互动显示周期”。

[0125] `selection_time_out_pts` 指定了在时元结束之前的一个时间。可以在从 $PTS(DS_n[ICS])$ 到 `selection_time_out_pts` 值所指定时间的周期内进行用户互动。该周期被称作“有效互动周期”。如上面所述,`selection_time_out_pts` 字段规定了这样一种周期的结束时间:该周期从互动显示的首次呈现开始并且在该周期期间可以将互动显示用于用户互动。`compostion_time_out_pts` 字段规定了这样一种周期:该周期从互动显示的首次呈现开始,并且在该周期期间时元内的互动显示保持有效并且因此被呈现。上述是对 `selection_time_out_pts` 和 `compostion_time_out_pts` 字段的描述。接下来将描述由 `selection_time_out_pts`、`compostion_time_out_pts` 以及 `user_time_out_duration` 字段所规定的多页菜单状态转换。

[0126] 图 12A 显示了基于 `selection_time_out_pts`、`compostion_time_out_pts` 以及 `user_time_out_duration` 字段的多页菜单状态转换。图 12A 显示了在使用弹出 U/I 情况下的多页菜单状态转换。该图中所示的多页菜单具有以下三种状态:无菜单显示、仅第一页、多页共存。

[0127] 仅第一页是其中仅呈现了多页菜单的第一页的状态。

[0128] 多页共存是其中除了第一页之外还呈现了第二页及其之后的页(即子页)的状态。

[0129] 箭头 jt1 代表了从无菜单显示状态到仅第一页状态的转换。该状态转换是由事件“Pop-Up_On”触发的。“Pop-Up_On”是在再现装置内响应于用户进行的菜单调用操作从而生成的事件。

[0130] 箭头 jt2 代表了从仅第一页状态到无菜单显示状态的转换。该状态转换是由事件“Pop-Up_Off”触发的。“Pop-Up_Off”是在再现装置内响应于用户进行的菜单移除操作从而生成的事件。

[0131] 箭头 jt3 和 jt4 分别代表了从仅第一页状态到多页共存状态的转换,以及从多页共存状态到仅第一页状态的转换。该状态转换是由到达 selection_time_out_pts 所规定的超时情况时生成的事件“ChangeButtonPage”触发的。“ChangeButtonPage”是在再现装置内响应于用户进行的切换菜单页的操作从而生成的事件。生成该事件的前提条件是用户进行切换菜单的操作。selection_time_out_pts 显示了强制性地使用于接收切换菜单页的用户操作的按钮变得无效的时间。因此,selection_time_out_pts 字段的描述实现了“仅第一页→多页共存→仅第一页”之间的状态转换。

[0132] 箭头 jt5 代表了从仅第一页状态到无菜单显示状态的转换。箭头 jt6 代表了从多页共存状态到无菜单显示状态的转换。jt6 所代表的状态转换是由 user_time_out_duration 的超时触发的。

[0133] 在该图中,从仅第一页状态到无菜单显示状态的状态转换(jt5)和从多页共存状态到无菜单显示状态的状态转换(jt6)是由 user_time_out_duration 字段规定的。也就是说,可以规定这样一种控制,从而当 AV 再现到达高潮场景时自动地移除屏幕上的菜单页。通过这种控制,可以避免菜单页干扰用户观看视频流再现。

[0134] 图 12B 显示了使用了总是开启 U/I 时的多页菜单状态转换。箭头 at1 代表了从无菜单显示状态到仅第一页状态的转换。这种状态转换是由事件“时元开始”触发的。事件“时元开始”显示了当前再现点到达了与 interactive_composition 结构相关的 PTS 所对应的点。如图中所示,随着视频流再现的进行,可以自动地呈现第一页而无需等待任何用户操作。

[0135] 箭头 at2 显示了从仅第一页状态到多页共存状态的转换,并且箭头 at3 显示了从多页共存状态到仅第一页状态的转换。这些状态转换也是由发生 selection_time_out_pts 所规定有效互动周期的超时情况时生成的事件“ChangeButtonPage”触发的。

[0136] 箭头 at4 代表了从多页共存状态到仅第一页状态的转换。该状态转换是由 user_time_out_duration 的超时触发的。通过该状态转换同样可以看出,当使用总是开启 U/I 时,基于发生 user_time_out_duration 的超时,状态转换是变为仅第一页状态而不是变为无菜单显示状态。

[0137] 箭头 at5 代表了从仅第一页状态到无菜单显示状态的转换。该状态转换是由 composition_time_out_pts 的超时触发的。注意到 composition_time_out_pts 显示了时元的结束(时元结束)。

[0138] 从图 12A 和 12B 可以明显地看出,多页菜单动作是由 interactive_composition 结构中的 selection_time_out_pts、compostion_time_out_pts 以及 user_time_out_duration 字段规定的。

[0139] 图 13-15 显示了根据上述控制所呈现的屏幕显示。图 13 显示了包含多页菜单的

显示情况。在这种情况下,开始时并未将多页菜单呈现在屏幕上。在当前再现点到达点 t1 时,呈现多页菜单的第一页,并且在到达 t2 时额外呈现一个子页。在当前再现点到达点 t3 时,移除屏幕上的子页而仅保留屏幕上的第一页。而在到达点 t4 时,移除第一页而使得屏幕上无菜单页显示。

[0140] 图 13 的底部是所描述的用于实现上述显示情况的 `interactive_composition` 结构。将 `PTS(DSn[ICS])` 设置为对应于点 t1 的值,将 `selection_time_out_pts` 设置为对应于点 t2 的值,将 `user_time_out_duration` 设置为对应于点 t3 的值,并且将 `compostion_time_out_pts` 设置为对应于点 t4 的值。通过这种设置,多页菜单就可以经历图 13 中所示的状态转换。

[0141] 接下来将参考一个具体例子,更加详细地描述 `selection_time_out_pts` 值的设置。

[0142] 图 14A-14C 显示了 `selection_time_out_pts` 所规定的一个显示例子。如图 14A 中所示,在点 t1 处,多页菜单中的第一页与视频流的一个画面叠加在一起。在点 t2 处呈现的画面是一种风景画面,并且在 AV 剪辑的故事情节中并不十分重要。因为该画面并未向用户提供有价值的内容,所以就设置 `selection_time_out_pts` 值以便在点 t2 处自动激活在第一页上呈现的一个按钮。因此,一个子页与图 14B 中所示的风景短片叠加在一起,从而在屏幕上出现如图 14C 中所示的合成画面。

[0143] 按照类似的方式,参考一个具体显示例子,描述如何设置 `user_time_out_duration` 值。图 15A-15D 显示了由 `user_time_out_duration` 所规定的一个显示例子。在该例子中,如图 15A 中所示,多页菜单处于多页共存状态,并且其中多个子页(第二页、第三页和第四页)呈现在屏幕上。在这种多页共存状态下,多页菜单的多个页与视频流中包含的一个画面叠加在一起。这里,将要在点 t3 处呈现的画面是如图 15B 中所示的人物画面。在这种情况下,如图 15C 中所示,与该画面叠加在一起的多个子页最终会覆盖该人物画面的大部分。因此,无法看到该人物画面的面部表情。为了避免这种干扰,将 `user_time_out_duration` 设置为对应于恰好位于点 t3 之前的一个时间点的值。通过这种设置,多页菜单就会经历从多页共存状态到仅第一页状态的状态转换。因此,在点 t3 处就会呈现如图 15D 中所示的合成画面。这里,仅在屏幕上呈现第一页,而不会过多地占据该人物画面。而且,因为还有第一页保留在屏幕上而不是移除多页菜单的每一页,所以不需要再次调用菜单。

[0144] 如上面所述,根据第一实施例,复用到图形流中的 ICS 包含了控制信息,其中该控制信息根据 AV 再现的进行规定了多页菜单的动作。这使得可以规定这样一种控制,以便在再现并不重要的场景(例如风景短片)时呈现很多页,而当 AV 再现接近高潮场景时移除屏幕上的页。通过这种控制,可以呈现菜单页而不会干扰作为主要内容的 AV 剪辑的再现。因此,用户可以集中精力观看 AV 再现,而同时可以享受多页菜单功能所带来的便利。

[0145] (第二实施例)

[0146] 第二实施例涉及一种用于呈现多页菜单的页的显示成分。根据第二实施例,显示成分用于在屏幕上呈现多个图形元素调用的按钮。另外,在引入菜单页之前和/或移除菜单页之后会在屏幕上呈现一种动画效果。

[0147] 图 16 显示了与多页菜单中多页的任意一页(第 x 页)相关的页信息的内部结构。如图中左侧部分所示,页信息(x)包含下述字段:“`page_id`”,显示了唯一地标识该

页 (x) 的标识符;“UO_mask_table”;“in_effects”,用于指定将页 (x) 引入到屏幕上时所呈现的动画效果;“out_effects”,用于指定将页 (x) 从屏幕上移除时所呈现的动画效果;“animation_frame_rate_code”,显示了在呈现用于页 (x) 的动画效果时需要遵循的动画帧速率;“default_selected_button_id_ref”;“default_activated_button_id_ref”;“pallet_id_ref”以及“button_info(1), (2)... (number_of_button-1)”中的每个与呈现在页 (x) 上的按钮之一相关。

[0148] “UO_mask_table”字段显示了允许和禁止对页 (x) 进行用户操作。如果该字段被设置为显示“禁止”的值,那么即使当页 (x) 处于活动状态,相应的对再现装置的用户操作也是无效的。

[0149] “default_selected_button_id_ref”字段显示了在开始呈现页 (x) 时究竟是动态地还是静态地确定将按钮选择为缺省按钮。当该字段被设置为“0xFF”时,动态地确定缺省选择按钮。在这样一种情况下,按照优先级解释再现装置的播放器状态寄存器 (PSR) 中的值,并且使 PSR 所示的按钮成为选中状态。另一方面,当该字段的值不为“0xFF”时,静态地确定缺省选择按钮。在这样一种情况下,使用“default_selected_button_id_ref”字段值覆盖 PRS,并且使该字段所指定的按钮成为选中状态。

[0150] “default_activated_button_id_ref”字段显示了在“selection_time_out_pts”字段值所规定的时间被自动激活的按钮。当“default_activated_button_id_ref”字段值被设置为“FF”时,在超时情况下处于选中状态的按钮被自动激活。另一方面,当该字段被设置为“00”时,不自动激活任何按钮。当该字段被设置为既不是“00”也不是“FF”的值时,该值被解释为静态地指定了要被自动激活的按钮的有效按钮序号。

[0151] “pallet_id_ref”字段显示了与在 CLUT 单元中被设置的调色板相关的 ID,另外将在下文对其进行描述。

[0152] 每个“button_info”结构是规定了要被呈现在页 (x) 上的按钮的信息。通过上述这些字段,规定了多页菜单的每一页。接下来,将描述“button_info”结构的内部结构。在接下来的描述中,用按钮 (i) 表示页 (x) 上多个按钮中的任意一个按钮。在图 16 中,箭头 cx1 指示了被摘出的要详细描述“button_info(i)”的内部结构。

[0153] 页上所呈现的每个按钮都有三种状态,即正常状态、选中状态以及激活状态。当按钮处于正常状态时,该按钮仅是被显示。当按钮处于选中状态时,由于用户操作导致该按钮目前正被关注,但是并未被激活。当按钮处于激活状态时,该按钮已经被激活。因为每个按钮都具有这三种状态,所以“button_info(i)”具有下述信息。

[0154] “button_id”字段显示了唯一地标识 interactive_composition 结构中的按钮 (i) 的一个值。

[0155] “button_numeric_select_value”字段显示了一个标记,该标记用于指示按钮 (i) 是否是在数值上可选择的。

[0156] “auto_action_flag”字段显示了按钮 (i) 是否被自动激活。当“auto_action_flag”字段被设置为“ON”(比特值“1”)时,当选择该按钮 (i),它并不转换为选中状态而是直接转换为激活状态。另一方面,当“auto_action_flag”字段被设置为“OFF”(比特值“0”)时,当选择按钮 (i) 时,它并不转换为激活状态而是转换为选中状态。

[0157] “button_horizontal_position”字段和“button_vertical_position”字段分别

指定了按钮 (i) 的顶部左侧像素在互动显示上的水平和垂直位置。

[0158] “neighbor_info”结构显示了当按钮 (i) 处于选中状态时响应于用户操作使得关注点沿上、下、左以及右方向移动,将要接收选中状态的按钮。“neighbor_info”结构包含“upper_button_id_ref”、“lower_button_id_ref”、“left_button_id_ref”以及“right_button_id_ref”字段。

[0159] “upper_button_id_ref”字段指定了当按钮 (i) 处于选中状态时并且若进行了命令关注点向上移动的用户操作的情况下,将要接收选中状态的按钮的按钮序号。当按下遥控器的 Move Up 键时,执行向上移动操作。如果该字段被设置为与按钮 (i) 相同的按钮序号,那么就忽略用户对 Move Up 键的操作。

[0160] 类似地,“lower_button_id_ref”、“left_button_id_ref”以及“right_button_id_ref”字段中的每个都指定了当按钮 (i) 处于选中状态时并且如果进行了命令关注点向下、向左或向右移动的用户操作情况下将要接收选中状态的按钮的按钮序号。当按下遥控器的 Move Down、Move Left 或 Move Right 键时,分别执行上述这些操作。如果这些字段被设置为与按钮 (i) 相同的按钮序号,那么就忽略用户对这些键的操作。

[0161] “normal_state_info”结构规定了按钮 (i) 的正常状态,并且包含“normal_start_object_id_ref”、“normal_end_object_id_ref”以及“normal_repeat_flag”字段。

[0162] “normal_start_object_id_ref”字段指定了被按顺序分配给 ODS 序列的多个 object_id 值中的第一个,其中该 ODS 序列用于通过动画的方式呈现按钮 (i) 的正常状态。

[0163] “normal_end_object_id_ref”字段指定了被按顺序分配给 ODS 序列的多个 object_id 值中的最后一个,其中该 ODS 序列用于通过动画的方式呈现按钮 (i) 的正常状态。如果“normal_end_object_id_ref”字段所指定的 ID 值与“normal_start_object_id_ref”所指定的 ID 值相同,那么将该 ID 所标识的图形对象的静态画面作为按钮 (i) 而呈现。

[0164] “normal_repeat_flag”字段指定了是否连续地重复处于正常状态的按钮 (i) 的动画。

[0165] “selected_state_info”结构规定了按钮 (i) 的选中状态。“selected_state_info”结构包含“selected_state_sound_id_ref”、“selected_start_object_id_ref”、“selected_end_object_id_ref”以及“selected_repeat_flag”字段。

[0166] “selected_state_sound_id_ref”字段指定了当按钮 (i) 被设置为选中状态时将要被作为点击声音从而再现的声音数据。该声音数据是由称作“sound.bdmv”的文件中包含的一条声音数据的“sound_id”指定的。当该字段被设置为“0xFF”时,不使任何声音数据与按钮 (i) 的选中状态相关联,从而当按钮 (i) 被改变为选中状态时不会再现任何点击声音。

[0167] “selected_start_object_id_ref”字段指定了被按顺序分配给 ODS 序列的多个 object_id 值中的第一个,其中该 ODS 序列用于通过动画的方式呈现按钮 (i) 的选中状态。

[0168] “selected_end_object_id_ref”字段指定了被按顺序分配给 ODS 序列的多个 object_id 值中的最后一个,其中该 ODS 序列用于通过动画的方式呈现按钮 (i) 的正常状态。如果“selected_end_object_id_ref”所指定的 ID 值与“selected_start_object_id_ref”所指定的 ID 值相同,那么将该 ID 所标识的图形对象的静态画面作为按钮 (i) 而呈现。

[0169] “selected_repeat_flag”字段指定了是否连续地重复处于选中状态的按钮 (i) 的

动画。如果“selected_end_object_id_ref”与“selected_start_object_id_ref”字段具有相同的值,那么就将“selected_repeat_flag”字段设置为值“00”。

[0170] “activated_state_info”结构规定了按钮 (i) 的激活状态并且包含“activated_state_sound_id_ref”以及“activated_start_object_id_ref”、“activated_end_object_id_ref”字段。

[0171] “activated_state_sound_id_ref”字段指定了当按钮 (i) 被设置为激活状态时将要被作为点击声音从而再现的声音数据。该声音数据是由“sound.bdmv”文件中包含的一条声音数据的“sound_id”指定的。当该字段被设置为“0xFF”时,不使任何声音数据与按钮 (i) 的激活状态相关联,从而当按钮 (i) 被改变为激活状态时不会再现任何点击声音。

[0172] “activated_start_object_id_ref”字段指定了被按顺序分配给 ODS 序列的多个 object_id 值中的第一个,其中该 ODS 序列用于通过动画的方式呈现按钮 (i) 的激活状态。

[0173] “activated_end_object_id_ref”字段指定了被按顺序分配给 ODS 序列的多个 object_id 值中的最后一个,其中 ODS 序列用于通过动画的方式呈现按钮 (i) 的激活状态。

[0174] “navigation_command”结构显示了当按钮 (i) 被激活时将要执行的导航命令。导航命令的一个代表性例子是 SetButtonPage 命令。SetButtonPage 命令使得再现装置显示多页菜单中希望得到的一页,并且在该页上多个按钮中的一个希望得到的按钮处于选中状态。通过使用这样的导航命令,内容创作者在进行创作时就可以容易地描述页转换。

[0175] 上面就是对“button_info”结构的描述。接下来将参考图 18,该图显示了规定按钮 0-A 到 0-D 的状态转换的 button_info 描述的例子,如图 17 中所示。在图 17 中,箭头 hh1 和 hh2 代表 button_info(1) 的 neighbor_info() 结构所规定的状态转换,其中 button_info(1) 结构与按钮 0-A 相关联。在 button_info(1) 中,lower_button_id_ref 字段被设置为指定了按钮 0-C 的值。因此,如果用户对 Move Down 键进行了操作 (图 17, up1) 而同时按钮 0-A 处于选中状态,那么按钮 0-C 就接收选中状态 (sj1)。在 button_info(1) 中的 right_button_id_ref 字段被设置为指定了按钮 0-B 的值。因此,如果用户对 Move Right 键进行了操作 (up2) 而同时按钮 0-A 处于选中状态,那么按钮 0-B 就接收选中状态 (sj2)。

[0176] 图 17 中的箭头 hh3 代表 button_info(2) 的 neighbor_info() 结构所规定的按钮状态转换,其中 button_info(2) 与按钮 0-C 相关联。在 button_info(2) 中,upper_button_id_ref 字段被设置为指定了按钮 0-A 的值。因此,如果用户对 Move Up 键进行了操作 (up3) 而同时按钮 0-C 处于选中状态,那么按钮 0-A 就被重新设置为选中状态 (sj3)。

[0177] 接下来将描述按钮 0-A、0-B、0-C 以及 0-D 的图形图像。配有图 18 中所示的 ICS 的 DS_n 包含对应于图 19 中所示的图形图像的 ODS11 到 ODS 19。button_info(0) 中的“normal_start_object_id_ref”和“normal_end_object_id_ref”分别指定了 ODS 11 和 ODS 13。因此,使用 ODS 11 到 ODS 13 的序列按照动画的形式呈现按钮 0-A 的正常状态。类似地,button_info(0) 中的“selected_start_object_id_ref”和“selected_end_object_id_ref”分别指定了 ODS 14 和 ODS 16。因此,使用 ODS 14 到 ODS 16 的序列按照动画的形式呈现按钮 0-A 的选中状态。通过这种设置,当进行用户操作从而将按钮 0-A 设置为选中状态时,起到按钮 0-A 作用的图形图像从使用 ODS 11-13 所呈现的按钮变为使用 ODS 14-16 所呈现的按钮。这里,如果 normal_state_info() 结构中的“normal_repeat_flag”字段被设置为值“1”,那么就按照图中所示“→ (A)”和“→ (A)”从而重复 ODS 11-13 的动画呈现。

类似地,如果 `selected_state_info()` 结构中的“`selected_repeat_flag`”字段被设置为值“1”,那么就按照图中所示“→ (B)”和“→ (B)”从而重复 ODS 14-16 的动画呈现。

[0178] 如上所述,可以按照动画形式呈现的多个 ODS 集合与按钮 O-A、O-B、O-C 以及 O-D 相关联。因此,通过用 ICS 参照 ODS,可以规定这样一种控制:起到按钮作用的人物图像响应于用户操作从而改变它的面部表情。

[0179] 上述是对 `button_info` 结构的描述。接下来将描述动画效果。如图 16 中所示,引入效果字段和退出效果字段分别指定了一种 `effect_sequence` 结构。图 20 显示了一种 `effect_sequence` 结构。如图中左侧所示, `effect_sequence` 结构包含 `window_info(0), (1), (2)... (number_of_windows-1)`; 以及 `effect_info(0), (1), (2)... (number_of_effects-1)`。

[0180] 通过按照固定的间隔更新显示成分从而呈现动画效果。每种 `effect_info` 结构都是一条规定了一种显示成分的信息。箭头 `ec1` 指示了将被摘出详细显示的 `effect_info` 的内部结构。如图中所示, `effect_info` 结构包含 `:effect_duration` 字段,显示了上述的固定间隔,即相关的显示成分将要被呈现的时间周期; `pallet_id_ref` 字段,指定了用于相关的显示成分的调色板;以及 `compositon_object(0), (1), (2)... (number_of_composition_object-1)`,指定了显示成分的详细情况。

[0181] 每个 `window_info` 结构都规定了图形平面上的一个窗口或一个区域,并且在该窗口或区域中绘制该显示成分。如箭头 `wc1` 所指示的, `window_info` 被摘出并被详细显示。如图中所示, `window_info` 包含下述字段:“`window_id`”,用于唯一地标识图形平面上的一个窗口;“`window_horizontal_position`”,用于指定该窗口的左上角像素的水平位置;“`window_vertical_position`”,用于指定该窗口的左上角像素的垂直位置;“`window_width`”,用于指定该窗口的宽度;以及“`window_height`”,用于指定该窗口的高度。

[0182] 接下来将描述可以设置在“`window_horizontal_position`”、“`window_vertical_position`”、“`window_width`”以及“`window_height`”字段的每个中的值。这些字段被设置为对应于二维图形平面内的坐标值,其中由 `video_height` 和 `video_width` 字段规定该图形平面的高度和宽度。

[0183] 因为“`window_horizontal_position`”字段显示了像素在图形平面上的水平地址,所以该字段的值范围为从“1”到 `video_width` 值。类似地,因为“`window_vertical_position`”显示了像素在图形平面上的垂直地址,所以该字段的值范围为从“1”到 `video_height` 值。

[0184] 因为“`window_width`”字段显示了窗口在图形平面上的宽度,所以该字段的值范围为从“1”到用 `video_width` 值减去 `widow_horizontal_position` 从而计算得到的值。类似地,因为“`window_height`”字段显示了窗口在图形平面上的高度,所以该字段的值范围为从“1”到用 `video_height` 值减去 `widow_vertical_position` 从而计算得到的值。

[0185] 如上所述, `window_info` 设置有规定了窗口在图形平面上的尺寸和位置的“`window_horizontal_position`”、“`window_vertical_position`”、“`window_width`”以及“`window_height`”字段。

[0186] 接下来将描述对窗口尺寸的限制,以便实现用于以每秒 24 帧的速率呈现图形显示成分的框架。该帧速率对应于视频流再现所采用的视频帧速率。通过该框架,按照与视

频流准确同步的方式呈现图形对象。为了实现该框架,必须按照视频帧速率清除和绘制该窗口。这里,要考虑到从对象缓存器到图形平面的所要求的传送速率。

[0187] 首先将检查对窗口尺寸的限制。用 R_c 表示从对象缓存器到图形平面的传送速率。在最坏情况下,需要以 R_c 的速率清除和绘制窗口。也就是说,需要以 R_c 的一半的速率 ($R_c/2$) 清除和绘制窗口。

[0188] 因此,需要满足下述等式:

[0189]

$$(\text{窗口尺寸}) \times (\text{帧速率}) \cong R_c/2$$

[0190] 如果帧速率是 24,那么通过下述等式表示 R_c :

$$[0191] \quad R_c = (\text{窗口尺寸}) \times 2 \times 24$$

[0192] 如果图形平面的像素总数是 1920×1080 ,并且每个像素的索引的位长度为 8 位,那么图形平面的总容量就是 2M 字节($\cong 1920 \times 1080 \times 8$)。

[0193] 如果 R_c 是 128Mbps,并且窗口尺寸是图形平面的 $1/A$,那么就满足下述等式:

$$[0194] \quad 128,000,000 = 1920 \times 1080 \times 1/A \times 8 \times 2 \times 24$$

[0195]

$$1/A \cong 0.16 (=128,000,000 / (1920 \times 1080 \times 8 \times 2 \times 24))$$

[0196] 因此,通过将窗口尺寸限制为整个图形平面的 16%,只要按照 128Mbps 的传送速率绘制动画效果,那么就可以保证动画效果与视频流之间的同步。

[0197] 例如,假设显示了清除和绘制窗口的速率的 effect_duration 为视频帧速率的 $1/2$ 或 $1/4$ 。在这种情况下,即时 R_c 仍保持不变,窗口尺寸可以为整个图形平面的 32% 或 64%。也就是说, effect_sequence 结构具有这样一种特性:窗口尺寸越大,所确定的 effect_duration 也就越长,而窗口尺寸越小, effect_duration 也就越短。上述就是对窗口尺寸的描述。接下来将描述窗口的位置。如上面所述,窗口的位置和尺寸在整个时元期间是固定的。

[0198] 由于下述原因,窗口的位置和尺寸在整个时元期间是固定的。如果窗口的位置或尺寸发生变化,那么就需要改变图形平面的写地址,而这又会导致系统开销。这种系统开销不可避免地会降低从对象缓存器到图形平面的传送速率。

[0199] 对于可以一次在一个窗口中绘制的图形对象的数量也有限制。对图形对象的数量进行限制是为了降低在转移解码后的图形对象时引起的系统开销。具体而言,这里所指的系统开销发生在设置图形对象的边缘部分的地址的时候。如果边缘部分的数量增多,该系统开销也会增大。

[0200] 如果不限制可以在一个窗口中显示的图形对象的数量,那么把图形对象转移到图形平面时会引发无限制的系统开销,而这会增加转移负载的波动。另一方面,如果将一个窗口中的图形对象的数量限制为两个,那么可以在假设系统开销的实例数量最多为四个的情况下设置传送速率 R_c 。因此,可以容易地确定传送速率 R_c 的最小标准。

[0201] 接下来考虑 effect_info 结构,并且根据窗口尺寸确定 effect_duration 。接下来将描述如何确定 effect_duration 。图 21A 示意性地显示了如何确定 effect_duration 。在该图中,在呈现了第 j 个显示成分后的多个视频帧之后,呈现第 $(j+1)$ 个显示成分。这些视频帧的数量就是 effect_duration 。为了计算 effect_duration ,必须知道为了呈现每个

显示成分所绘制的数据的总量。这里, `effect_sequence` 设置有多个窗口, 并且这些窗口的数量由 `number_of_widows` 字段的值指示。因此, 为了呈现第 j 个显示成分所绘制的数据的总量就等于从窗口 (0) 到窗口 (`number_of_widows-1`) 的总量, 如大括号 `br` 所指示的那样。

[0202] 如上面所述, 第 j 个显示成分包含从窗口 (0) 到窗口 (`number_of_widows-1`)。因此, 用这些窗口的总数据量除以再现装置的传送速率 (128Mbps) 然后再乘以 PTS 的时间精度 (90KHz), 就可以得到用 90KHz 表示的结果。得到的值是呈现 `effect_sequence` 的第 j 个显示成分所用的时间长度。图 21B 显示了计算 `effect_duration` 的等式。

[0203] 在上面的描述中应该注意到, 运算符 “`ceil()`” 表示一种除法, 通过该除法返回一个被四舍五入得到的整数值。

[0204] 图 22 显示了引入效果动画的一个具体例子。引入效果动画呈现这样一种画面, 使得屏幕上人物图像的掌心处出现一道闪光, 并且在该闪光消失后逐渐出现一个菜单。当在屏幕上完全显示该菜单时 (`effect(4)`), 引入效果内的图形数据量是最大的。因此, 根据 `effect(4)` 的数据量确定窗口的 `window_width` 和 `window_height`。假设这里所确定的窗口尺寸大约为整个图形平面的 50%。该窗口尺寸几乎是上述窗口尺寸, 即图形平面的 16% 的三倍。因此, 该引入效果的 `effect_duration` 被确定为较长, 以便可以按照 128Mbps 的速率对显示成分进行及时的更新。因此, 将 `effect(0)`、(1)、(2) 和 (3) 的每个 `effect_duration` 设置为相对较长。根据这种设置, 为每三个视频帧绘制一个显示成分, 从而可以确保无缝地呈现引入效果动画。相同的方法也适用于退出效果动画。图 23 显示了退出效果动画, 并且根据上述方式确定该退出效果动画的 `window_width`、`window_height` 以及 `effect_duration` 字段值。

[0205] 如上面所述, 根据第二实施例, `window_info` 规定了图形平面上用于绘制的一个区域, 并且 `effect_info` 规定了一种持续时间, 并且在该持续时间的间隔对该绘制区域进行更新。因此, 当该绘制区域较大时, 可以调整该持续时间从而使其较长; 当该绘制区域较小时, 可以调整该持续时间从而使其较短。通过这种设置, 只要再现装置按照固定的传送速率传送图形数据, 就可以确保无缝地呈现动画效果。

[0206] (第三实施例)

[0207] 本发明的第三实施例涉及对动画效果的呈现的改进, 例如滚动、擦入 / 擦出 (`wipe-in/out`) 以及切入 / 切出 (`Cut-In/Out`)。用于实现这些动画效果的结构位于图 24 所示的 `composition_object` 结构中。图 24 详细显示了任意一个 `composition_object(i)` 的内部结构。如图中所示, `composition_object(i)` 包含下述字段: “`object_id_ref`”、“`window_id_ref`”、“`object_cropped_flag`”、“`composition_object_horizontal_position`”、“`composition_object_vertical_position`” 以及 “`cropping_rectangle_info(1), (2)... (n)`”。

[0208] “`object_id_ref`” 字段显示了图形对象标识符的参考值 (`object_id`)。该参考值对应于标识了图形对象的 `object_id`, 其中该图形对象用于根据 `composition_object(i)` 从而生成显示成分。

[0209] “`window_id_ref`” 字段显示了窗口标识符的参考值 (`window_id`)。该参考值对应于标识窗口的 `window_id`, 在该窗口中绘制了根据 `composition_object(i)` 的显示成分。

[0210] “`object_cropped_flag`” 显示了一种标记, 该标记指示了是否呈现对象缓存器中

被裁切的图形对象。当该字段设置为“1”时,呈现对象缓存器中被裁切的图形对象。另一方面,当该字段设置为“0”时,不呈现对象缓存器中被裁切的图形对象。

[0211] “composition_object_horizontal_position”字段显示了图形对象的左上角像素在图形平面上的水平位置。

[0212] “composition_object_vertical_position”字段显示了图形对象的左上角像素在图形平面上的垂直位置。

[0213] “cropping_rectangle”结构在“object_cropped_flag”字段设置为“1”的情况下是有效的。在图 24 中,箭头 wd2 指示了被摘出将要详细描述的结构 cropping_rectangle 的内部结构。如图中所示, cropping_rectangle 结构包含下述字段:“object_cropping_horizontal_position”、“object_cropping_vertical_position”、“object_cropping_width”以及“object_cropping_height”。

[0214] “object_cropping_horizontal_position”字段显示了裁切矩形的左上角的水平位置。该裁切矩形规定了要被裁切的图形对象的区域。

[0215] “object_cropping_vertical_position”字段显示了裁切矩形的左上角的垂直位置。

[0216] “object_cropping_width”字段显示了裁切矩形的宽度。

[0217] “object_cropping_height”字段显示了裁切矩形的高度。

[0218] 上述是对 composition_object 的结构描述。接下来将描述 composition_object 的结构的一个具体例子。

[0219] 图 25 显示了呈现一种右滚动动画效果的引入效果的一个具体例子。引入效果动画的该例子呈现了这样一种画面,使得一个语言选择菜单逐渐从屏幕的右边缘出现。为了实现该引入效果动画, composition_object(0)、composition_object(1)、composition_object(2) 以及 composition_object(3) 分别被分配到点 t0、t1、t2 以及 t3。而且,相关于每个 DS 的 ICS 和 effect_info 结构被规定为包含下述 window_info 结构和 composition_object 结构。

[0220] 接下来将描述每个 composition_object 结构的设置。图 26-29 中的每个都显示了 composition_object 结构的一个示例性设置。图 26 显示了 composition_object(0) 结构的一个例子。

[0221] 在该图中, window_horizontal_position 和 window_vertical_position 字段指定了窗口的左上角像素 LPO 在图形平面上的坐标。window_width 和 window_height 字段指定了该窗口的宽度和高度。

[0222] 如图中所示,“object_cropping_horizontal_position”和“object_cropping_vertical_position”字段指定了一个参考点 ST1,并且该参考点用于确定要被裁切的图形对象的区域。根据其原点位于对象缓存器中图形对象的左上角的坐标系从而对参考点 ST1 进行寻址。从参考点 ST1 开始并且具有由“object_cropping_width”和“object_cropping_height”字段所指定的长度和宽度的区域被确定为裁切矩形(图中的实线框)。将图形对象中被裁切的部分绘制到图形平面上这样一个位置:该位置的左上角像素 LPO 是由“composition_object_horizontal_position”和“composition_object_vertical_position”字段指定的。在该例子中,语言选择菜单的右侧大约 1/4 被绘制在图形平面的窗

口内。因此,语言选择菜单的右侧四分之一作为叠加在视频上的合成图形从而被呈现。

[0223] 图 27、28 和 29 分别显示了 composition_object 结构 (1)、composition_object 结构 (2) 以及 composition_object 结构 (3) 的例子。每个图中所示的 window_info 结构都与图 26 中所示的该结构相同。因此,这里就不再重复描述。但是,图 27 中所示的 composition_object (1) 结构与图 26 中所示的结构不同。在图 27 中,“object_cropping_horizontal_position”和“object_cropping_vertical_position”字段指定了存储在对象缓存器中语言选择菜单的大约右侧一半(而不是 1/4)的左上角像素的坐标。“object_cropping_width”和“object_cropping_height”字段指定了语言选择菜单的右半边的高度和宽度。类似地,在图 28 中,composition_object (2) 结构的各个字段一起指定了语言选择菜单的右侧大约 3/4。在图 29 中,composition_object (3) 结构的各个字段一起指定了语言选择菜单的整个部分。通过将图 27、28 和 29 中所示的图形对象的裁切部分绘制到图形平面,可以将语言选择菜单的右半边、右侧 3/4 以及整个部分叠加到视频上。

[0224] 如上面所述,根据本发明的第三实施例,通过 composition_object 结构,可以容易地规定所希望得到的动画效果。例如,可以容易地规定按照这样一种方式绘制存储于对象缓存器中的图形对象,使得该图形对象逐渐出现在屏幕上或者逐渐从屏幕上消失。

[0225] (第四实施例)

[0226] 本发明的第四实施例涉及通过优化 DS 内的 ODS 的顺序从而实现的一种改进。如上面所述,通过 interactive_composition 来参照 DS 内的 ODS,以便呈现引入效果或退出效果动画或者按钮的一种状态。DS 内的 ODS 的顺序是根据这些 ODS 是用于引入效果动画、退出效果动画还是按钮状态而确定的。

[0227] 具体而言,DS 内的 ODS 被分组为 (1) 用于呈现 PAGE[0] 的引入效果动画的 ODS、(2) 用于呈现按钮的正常状态的 ODS、(3) 用于呈现按钮的选中状态的 ODS、(4) 用于呈现按钮的激活状态的 ODS、(5) 用于呈现 PAGE[0] 的引入效果以及后面的页 [1] 的引入效果和退出效果动画的 ODS。也就是说,相关于同一种呈现的 ODS 被分在相同的组中。相关于引入效果的 ODS 组被称作“引入效果组”。相关于每种按钮状态的 ODS 组被称作“按钮状态组”。相关于 PAGE[0] 的退出效果以及后面的页 [1] 的引入效果和退出效果的 ODS 组被称作“其余组”。

[0228] 按照下述顺序排列这些 ODS 组:引入效果组→正常状态组→选中状态组→激活状态组→其余组。通过这种方式,根据与 ODS 相关的呈现目标从而确定这些 DS 内的 ODS 的顺序。

[0229] 接下来将具体描述 DS_n 内的 ODS 的顺序。图 30 显示了 DS_n 内的 ODS 的顺序。

[0230] 在该图中,ODS₁-ODS_g 属于与 PAGE[0] 的引入效果相关的 ODS 组(引入效果组)。

[0231] ODS_{g+1}-ODS_h 是与按钮的正常状态相关的 ODS 组(正常状态的 ODS)。

[0232] ODS_{h+1}-ODS_i 属于与按钮的选中状态相关的 ODS 组(选中状态的 ODS)。

[0233] ODS_{i+1}-ODS_j 属于与按钮的激活状态相关的 ODS 组(激活状态的 ODS)。

[0234] ODS_{j+1}-ODS_k 属于与 PAGE[0] 的退出效果相关的 ODS 以及页 [1] 及其后面的页的引入效果和退出效果相关的 ODS 组(其余组)。

[0235] 按照下述顺序排列这些 ODS 组:引入效果组→正常状态组→选中状态组→激活状态组→其余组。确定这样一种顺序使得可以首先读取互动显示的第一呈现所需的 ODS,而稍

后再读取仅在显示更新之后才需要的 ODS。

[0236] 接下来将描述当存在多引用时 ODS 的顺序。这里所用的术语“多引用”意味着 ICS 中的两条或多条 `normal_state_info`、`selected_state_info` 和 `activated_state_info` 都引用了同一个 `object_id`。例如,通过多引用,用于绘制处于正常状态的按钮的特定图形对象通常会被用于绘制处于选中状态的另一个按钮。也就是说,共享该图形对象,以便降低 ODS 的数量。这里,会出现这样一种问题:具有多引用的 ODS 究竟属于哪个按钮状态组。

[0237] 具体而言,当一个 ODS 既与处于正常状态的一个按钮相关,同时又与处于选中状态的另一个按钮相关时,需要确定该 ODS 究竟是属于正常状态的按钮状态组,还是属于选中状态的按钮状态组。

[0238] 在这种情况下,将 ODS 放入在 ICS 中首先出现的按钮状态组中。

[0239] 例如,如果正常状态组和选中状态组都引用同一个 ODS,那么就将该 ODS 放入正常状态组 (N-ODS),而不将其放入选中状态组 (S-ODS)。同样,如果选中状态组和激活状态组都同时引用一个 ODS,那么就将该 ODS 放入选中状态组 (S-ODS),而不将其放入激活状态组 (A-ODS)。通过这种方式,对于被两个或多个不同的 `state_info` 结构所引用的每个 ODS,该 ODS 仅在 DS 中出现一次,并且被放入按钮状态组的第一个中。

[0240] 上述也适用于与动画效果相关的 ODS。如果一个图形对象被 `PAGE[0]` 的引入效果和退出效果所共用,那么该图形对象仅在 DS 中出现一次,并且被放入引入效果组中。类似地,如果一个图形对象被 `PAGE[0]` 的引入效果和后面的页 `[1]` 的引入效果或退出效果所共用,那么该图形对象仅在 DS 中出现一次,并且被放入引入效果组中。上面描述了当存在对 ODS 的多个引用时 ODS 的顺序。

[0241] 因为可能存在如上所述的对 ODS 的多个引用,所以 DS_n 可能并不包含上述所有 ODS 组。例如,由于多引用, DS_n 可能并不包含选中状态组 ODS。而且, DS_n 可能并不包含引入效果组 ODS。这也是因为呈现互动显示时并不一定必须具有引入效果。

[0242] 接下来将描述选中状态的按钮状态组内的 ODS (S-ODS) 的顺序。在这些 S-ODS 中,应该将哪一个 ODS 放在开始位置依赖于动态地还是静态地确定了缺省选择按钮。如果 `interactive_composition` 中 `default_selected_button_id_ref` 字段设置为除“00”之外的其它有效值,那么就静态地确定缺省选择按钮。该有效值指定了将要被选为缺省的按钮。如果 `default_selected_button_id_ref` 字段值是有效的并且相关于所指定的缺省选择按钮的 ODS 并不位于 N-ODS 中,那么就将这些 ODS 放在 S-ODS 的开始位置。

[0243] 另一方面,如果 `default_selected_button_id_ref` 字段设置为“00”,那么就根据再现装置的状态从而动态地选择一个不同的按钮作为缺省。

[0244] 例如,在复用有 DS 的 AV 剪辑起到多个再现路径的合并点的作用的情况下,`default_selected_button_id_ref` 字段设置为值“00”。在这种情况下,需要根据采用了哪条再现路径从而选择一个不同的按钮作为缺省按钮。因此,不需要将特定的 ODS 放在 S-ODS 的开始位置。

[0245] 图 31 显示了在 `default_selected_button_id_ref` 字段设置为“00”和设置为指定了按钮 B 的有效值的情况下,S-ODS 内的 ODS 的顺序之间的区别。在该图中,箭头 `ss1` 指示了 `default_selected_button_id_ref` 字段指定了用于标识按钮 B 的 `button_id` 的情况下,S-ODS 内的 ODS 的顺序。箭头 `ss2` 指示了 `default_selected_button_id_ref` 字段设置为

“00”的情况下，S-ODS 内的 ODS 的顺序。如图中所示，在 default_selected_button_id_ref 字段指定按钮 B 的情况下，将相关于选中状态按钮 B 的 ODS（按钮 B 的 ODS）放在 S-ODS 的开始位置，其后再放置相关于其它按钮的 ODS（按钮 A、C 和 D 的 ODS）。另一方面，在 default_selected_button_id_ref 字段设置为“00”的情况下，将相关于选中状态按钮 A 的 ODS 放在 S-ODS 的开始位置。如上面所述，根据 default_selected_button_id_ref 字段值是否被赋值，S-ODS 内的 ODS 的顺序是不同的。

[0246] 根据该第四实施例，按照 ODS 在 DS 内出现的顺序从而排列 DS 内的 ODS。因此，可以平滑地呈现包含多个图形元素的显示成分。

[0247] （第五实施例）

[0248] 在第一实施例中，已经描述了如何将 DS 分配到 AV 剪辑再现的时间轴。在本发明的第五实施例中，将要描述如何设置 PTS 和 DTS 值。

[0249] 首先，描述基于 ODS 内的 DTS 和 PTS 进行同步控制的机制。

[0250] DTS 显示了需要开始对相关的 ODS 进行解码的时间，其中时间精度为 90KHz。PTS 显示了完成该解码的截止时间。

[0251] ODS 的解码无法在瞬间就完成，而是需要一定的持续时间。为了明确地显示 ODS 的解码的开始和结束时间，使用 DTS 和 PTS 显示了解码开始时间和解码截止时间。

[0252] PTS 值显示了这样一种截止时间：其中需要在该时间之前完成相关 ODS 的解码，并且需要在再现装置的对象缓存器中获得所得到的未压缩的图形对象。

[0253] 属于 DS_n 的任意一个 ODS_j 的解码开始时间是由 $DTS(DS_n[ODS_j])$ 所示，并且时间精度为 90KHz。因此， ODS_j 的解码截止时间是由 $DTS(DS_n[ODS_j])$ 值与进行解码所需的最长时间之和确定的。

[0254] 这里，用 $SIZE(ODS_n[ODS_j])$ 表示解压缩之后的 ODS_j 的大小，并且用 Rd 表示 ODS_j 的解码速率，那么解码所需的最长时间就是 $ceil(SIZE(DS_n[ODS_j])/Rd)$ 。

[0255] 通过将该最长时间转换为 90KHz 的精度并且将结果与 ODS_j 的 DTS 相加，就可以计算得到精度为 90KHz 的由 PTS 表示的解码截止时间。

[0256] 可以用下述等式表示属于 DS_n 的 ODS_j 的 PTS：

$$[0257] \quad PTS(DS_n[ODS_j]) = DTS(DS_n[ODS_j]) + 90,000 \times ceil(SIZE(DS_n[ODS_j])/Rd)$$

[0258] 另外，两个相邻的 ODS (ODS_j 和 ODS_{j+1}) 之间的关系需要满足下述等式：

$$[0259] \quad PTS(DS_n[ODS_j]) \leq DTS(DS_n[ODS_{j+1}])$$

[0260] 上述就是对 ODS 的 PTS 和 DTS 的描述。接下来将描述 ICS 内的 PTS。在 DS_n 内的第一个 ODS 的解码开始时间之前（即在 $DTS(DS_n[ODS_1])$ 所示的时间之前）并且在 DS_n 的首次呈现变为有效之前（即 $PTS(DS_n[PDS_1])$ 所示的时间），需要将 ICS 加载到再现装置的成分缓存器。因此，需要满足下述等式：

$$[0261] \quad DTS(DS_n[ICS]) \leq DTS(DS_n[ODS_1])$$

$$[0262] \quad DTS(DS_n[ICS]) \leq PTS(DS_n[PDS_1])$$

[0263] 接下来将描述 DS_n 的 ICS 内的 PTS。PTS($DS_n[ICS]$) 值需要满足下述等式：

$$[0264] \quad PTS(DS_n[ICS]) \geq PTS(DS_n[ICS]) + DECODEDURATION + TRANSFERDURATION$$

[0265] DECODEDURATION 显示了对 DS_n (ICS) 的首次呈现所需的图形对象进行解码所需的时间。解码开始时间等于最早时候的 DTS(ICS) 值。这里，用 LASTODSPTS 表示与一种图形

对象相关的 PTS, 其中该图形对象在 DS_n (ICS) 的首次呈现所需的图形对象中是最晚完成解码的一个, 因此 $DECODEDURATION$ 等于 $LASTODSPTS(DS_n) - DTS(DS_n[ICS])$ 。

[0266] 根据 `default_selected_button_id_ref` 设置和 DS 内 ODS 的顺序, $LASTODSPTS$ 值会具有不同的值。图 32A、32B、33A、33B 和 33C 显示了不同的 ODS 顺序, 以说明使用 ODS 的 DTS 所进行的 $LASTODSPTS$ 设置。

[0267] 图 32A 显示了其中静态地确定缺省选择按钮的情况, 即其中 `default_selected_button_id_ref` 字段被设置为非零值的情况。在这种情况下, $LASTODSPTS$ 具有 S -ODS 内第一个 ODS 的 PTS 的值 (即 $PTS(DS_n[ODS_{n+1}])$ 的值)。在该图中, ODS_{n+1} 被包含在实线框中。

[0268] 图 32B 显示了其中动态地确定缺省选择按钮的情况, 即其中 `default_selected_button_id_ref` 被设置为“00”的情况。在这种情况下, $LASTODSPTS$ 具有 S -ODS 中最后一个 ODS 的 PTS 的值 (即 $PTS(DS_n[ODS_1])$ 的值)。在该图中, ODS_1 被包含在实线框中。

[0269] 图 33A 显示了不包含与选中状态相关的 ODS 的 DS 。在这种情况下, $LASTODSPTS$ 具有 N -ODS 中最后一个 ODS 的 PTS 的值 (即 $PTS(DS_n[ODS_n])$ 的值)。在该图中, ODS_n 被包含在实线框中。

[0270] 图 33B 显示了不包含与正常状态相关的 ODS 的 DS 。在这种情况下, $LASTODSPTS$ 具有引入效果 ODS 中最后一个 ODS 的 PTS 的值 (即 $PTS(DS_n[ODS_g])$ 的值)。在该图中, ODS_g 被包含在实线框中。

[0271] 图 33C 显示了不包含 ODS 的 DS 。在这种情况下, $LASTODSPTS$ 具有 DS_n 的 ICS 中指定的 DTS 值 (即 $DTS(DS_n[ICS])$ 的值)。在该图中, ICS 被包含在实线框中。

[0272] 注意到上面描述适用于 DS_n 并不是时元开始 DS 的情况。如果 DS_n 是时元开始 DS , 那么就采用不同的方法。

[0273] 在时元开始 DS 的情况下, 可能会清除整个图形平面。清除图形平面所需的时间可能会比通过 $LASTODSPTS(DS_n) - DTS(DS_n[ICS])$ 计算得到的时间要长。根据 BD-ROM 的目标解码器模型, 由不同的处理单元执行图形解码和图形平面清除 (图形处理器执行解码)。因此, 图形解码和平面清除并行地执行。即使已经完成了对呈现第一显示成分所需的图形数据进行的解码, 除非已经清除了图形平面, 否则还是无法将图形数据绘制到图形平面。因此, $DECODEDURATION$ 需要设置为等于平面清除持续时间的值。假设图形平面的宽度和高度分别等于 `video_width` 和 `video_height` 字段值。而且, 对象缓存器和图形平面之间的传送速率为 128Mbps。为了清除整个图形平面, 再现装置需要按照 128Mbps 的传送速率重写尺寸为 $8 \times video_width \times video_height$ 的图形平面。按照 90KHz 表示进行该重写所需的持续时间, 那么可以按照下述等式计算清除图形平面所需的时间 (称作“ $PLANECLARTIME$ ”):

[0274] $PLANECLARTIME = \text{ceil}((90000 \times 8 \times DS_n[ICS].video_width \times DS_n[ICS].video_height) / 128,000,000)$

[0275] 图 34A 显示了通过将 $LASTODSPTS(DS_n) - DTS(DS_n[ICS])$ 计算得到的值和 $PLANECLARTIME$ 值获取 $DECODEDURATION$ 值的算法。在该算法中, 将 $LASTODSPTS(DS_n) - DTS(DS_n[ICS])$ 值和 $PLANECLARTIME$ 值中较大的一个确定为 $DECODEDURATION$ 值。

[0276] 如图中所示, 如果 DS_n (ICS) 的 `composition_state` 字段指示了 DS_n 并不是时元开始 DS (`if(DS_n(ICS).composition_state == EPOCH_STATE) else`), 那么就

将 $\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])$ 值确定为 DECODEDURATION 值 ($\text{return}(\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])))$ 。另一方面, 如果 $\text{DS}_n(\text{ICS})$ 的 composition_state 字段指示了 DS_n 是时元开始 $\text{DS}(\text{if}(\text{DS}_n(\text{ICS}).\text{composition_state} == \text{EPOCH_START}))$, 那么将 $\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])$ 值和 PLANECLARTIME 值中较大的一个确定为 DECODEDURATION 值 ($\text{return}(\text{max}(\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])))$)。

[0277] 图 34B 和 34C 显示了根据 PLANECLARTIME 值是否大于 $\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])$ 值从而进行的 DECODEDURATION 设置的例子。如果 PLANECLARTIME 值大于 $\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])$ 值, 那么就确定 PLANECLARTIME 为 DECODEDURATION 值, 如图 34B 中所示。另一方面, 如果 PLANECLARTIME 值小于 $\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])$ 值, 那么就确定 $\text{LASTODSPTS}(\text{DS}_n) - \text{DTS}(\text{DS}_n[\text{ICS}])$ 值为 DECODEDURATION 值, 如图 34C 中所示。

[0278] TRANSFERDURATION 显示了将呈现第一显示成分所需的图形数据从对象缓存器传送到图形平面所需的时间。依赖于是否为 DS 提供了引入效果以及究竟是静态地还是动态地确定缺省选择按钮, DS 内的第一显示成分是不同的。因此, 根据第一显示成分, TRANSFERDURATION 也是有不同的。如果第一显示成分相关于引入效果, 那么将传送用于呈现该引入效果所需的未压缩图形的时间确定为 TRANSFERDURATION 。

[0279] 这里, $\text{DS}_n[\text{ICS}]$ 所规定的引入效果是用于引入 $\text{PAGE}[0]$ 并且绘制到多个窗口中, 其中这些窗口的数量由 number_of_window 字段值指示。因此, 引入效果的第一显示成分所需的图形数据的总量等于窗口 (0) 到窗口 ($\text{number_of_window}-1$) 的总量。因此, 为了计算传送用于呈现引入效果序列内第一显示成分所需的图形数据的时间 (EFFECTTD), 就用总窗口大小除以再现装置的传送速率 (128Mbps)。为了按照 PTS 的时间精度 (90KHz) 表示结果, 进一步将该结果乘以 90KHz。图 35A 中显示了计算 $\text{EFFECTTD}(\text{DS}_n)$ 的等式。

[0280] 另一方面, 如果 DS_n 不具有任何引入效果, 那么用于呈现 $\text{DS}_n[\text{ICS}]$ 的第一显示成分就是多页菜单的 $\text{PAGE}[0]$ 。如果是静态地指定了缺省选择按钮, 那么一旦用于缺省选择按钮的选中状态和其它按钮的正常状态所需的未压缩数据从对象缓存器传送到图形平面, 那么就立即呈现多页菜单的 $\text{PAGE}[0]$ 。

[0281] 这里, 相关于 $\text{DS}_n[\text{ICS}]$ 的多页菜单的 $\text{PAGE}[0]$ 包含多个按钮, 其中这些按钮的数量由 number_of_button 字段值指示。因此, 按照下述方式计算呈现第一显示成分所需的数据量。首先, 计算相关于页面上所有按钮的正常状态的图形数据的总量, 即从 (0) 到 ($\text{number_of_button}-1$) 的 $\text{DS}_n[\text{ICS}].(\text{ICS}).\text{Page}(0).\text{Button}$ 所指定的。然后, 使计算得到的总数据量与相关于 $\text{PAGE}[0]$ 的缺省选择按钮的选中状态的图形数据量 ($\text{SBSIZE}(\text{DS}_n, \text{DS}_n[\text{ICS}].\text{PAGE}(0).\text{default_selected_button_id_ref}))$ 相加。而且, 从得到的结果中减去相关于缺省选择按钮的正常状态的图形数据量 ($\text{NBSIZE}(\text{DS}_n, \text{DS}_n[\text{ICS}].\text{PAGE}(0).\text{default_selected_button_id_ref}))$, 从而获得呈现第一显示成分所需的数据量。

[0282] 用计算得到的呈现第一显示成分所需的数据量除以再现装置的传送速率 (128Mbps), 然后乘以 90KHz, 从而可以按照 PTS 的时间精度 (90KHz) 表示结果。然后, 计算呈现第一页所需的传送持续时间。图 35B 显示了在不具有引入效果并且静态地确定缺省选择按钮情况下用于计算 $\text{PAGDEFAULTTD}(\text{DS}_n)$ 的等式。

[0283] 在动态地确定缺省选择按钮情况下, 不能应用图 35B 中所示的等式, 这是因

为并不知道 PAGE[0] 上的哪个按钮被选择为缺省按钮。因此,需要计算最坏情况下的 TRANSFERDURATION。该最坏情况是这样一种情况:将 $DS_n[ICS].PAGE(0)$ 所规定的按钮中具有最大尺寸的按钮选择为缺省按钮。在 $DS_n[ICS].PAGE(0)$ 的按钮中,用 $LRG\{button: button \in DS_n[ICS].PAGE(0).Button\}$ 表示具有最尺寸的按钮。

[0284] 然后,按照下述方式计算上述最坏情况下需要被传送的数据量(最大数据量)。首先,计算 PAGE[0] 上所有按钮的总量(从 (0) 到 (number_of_button-1) 的 Page(0).Button)。使按钮 LRG 的大小 ($B\text{SIZE}(DS_n, LRG\{button: button \in DS_n[ICS].PAGE(0).Button\})$) 与计算得到的总大小相加。而且,从得到的结果中减去相关于按钮 LRG 的正常状态的图形数据量 ($N\text{BSIZE}(DS_n, LRG\{button: button \in DS_n[ICS].PAGE(0).Button\})$), 从而获得最大数据量。

[0285] 用按照上述方式计算得到的最大数据量除以再现装置的传送速率 (128Mbps), 然后乘以 90KHz, 从而可以按照 PTS 的时间精度 (90KHz) 表示结果。然后,计算呈现多页菜单的第一页所需的传送持续时间。图 35C 显示了在不具有引入效果并且动态地确定缺省选择按钮情况下用于计算 $PAGENODEFAULTTD(DS_n)$ 的等式。

[0286] 图 36 显示了用于有选择地确定 $EFFECTTD$ 、 $PAGEDEFAULTTD$ 、以及 $PAGENODEFAULTTD$ 中的一个为 TRANSFERDURATION 的算法。如图中所示,如果至少提供了一个引入效果 ($if(DS_n[ICS].PAGE[0].IN-EFFECTS.number_of_effect \neq 0)$), 那么就确定 $EFFECTTD$ 为 TRANSFERDURATION($return\ EFFECTTD(DS_n)$)。如果引入 PAGE[0] 时未提供引入效果并且未静态地指定缺省选择按钮 ($DS_n[ICS].PAGE(0).default_selected_button_id_ref = 0xFFFF$), 那么就确定 $PAGEDEFAULTTD$ 为 TRANSFERDURATION($return\ PAGENODEFAULTTD(DS_n)$)。如果引入 PAGE[0] 时未提供引入效果并且静态地确定缺省选择按钮,那么就确定 $PAGEDEFAULTTD$ 为 TRANSFERDURATION($return\ PAGEDEFAULTTD(DS_n)$)。

[0287] 如上面所述,根据本发明的第五实施例,基于 ODS 的数据量从而优化地确定了 ICS 的 DTS 和 PTS 值。因此,可以平滑地进行互动控制而不会出现延迟。

[0288] (第六实施例)

[0289] 本发明的第六实施例涉及用于再现上面第一到第五实施例中所述的 BD-ROM 的再现装置。图 37 显示了该再现装置的内部结构。根据第六实施例的再现装置是基于图 37 中所示的内部结构从而实现其工业制造。该再现装置主要包含两个部分,一个是系统 LSI,另一个驱动设备。通过将这些部分安装到装置的机壳和基板上可以实现其工业制造。系统 LSI 是一种包含多种用于实现再现装置的功能的处理单元的集成电路。

[0290] 按照上述方式制造的再现装置包含 BD 驱动器 1、读取缓存器 2、解复用器 (De-MUX) 3、视频解码器 4、视频平面 5、P 图形解码器 6、呈现平面 7、加法器 8、I 图形解码器 9、互动图形平面 10、加法器 11、音频解码器 13、CLUT(色彩查找表)单元 14 和 15、PRS 集合 16、系统时钟 17、user_timeout 定时器 18、selection_timeout 定时器 19、composition_timeout 定时器 20、effect_duration 定时器 21、操作接收单元 22 以及状态控制单元 23。

[0291] BD 驱动器 1 执行 BD-ROM 的加载、读取和弹出。BD 驱动器 1 对 BD-ROM 100 进行访问。

[0292] 读取缓存器 2 是一种 FIFO(先进先出)存储器。因此,从 BD-ROM 中读取的 TS 包被按照它们到达读取缓存器 2 的顺序从缓存器 2 中移除。

[0293] De-MUX 3从读取缓存器2读取TS包并且将它们转换为PES包。对于得到的PES包中那些具有预先确定的流PID的PES包来说,它们被输出到视频解码器4、音频解码器13、P图形解码器6和I图形解码器9中的一个。

[0294] 视频解码器4对从De-MUX 3接收到的PES包进行解码,并且将得到的未压缩画面数据绘制到视频平面5。

[0295] 视频平面5是用于存储未压缩画面数据的平面存储器。名词“平面”是指用于存储一个屏幕的像素数据的再现装置的存储区域。通过提供多个平面,再现装置可以将存储的数据按照逐个像素的方式添加到每个平面上,从而产生视频输出,由此将多个图形画面叠加在一起形成一个合成画面。视频平面5的分辨率为 1920×1080 。按照用YUV值表示的16位像素数据的方式将画面数据存储在视频平面5上。

[0296] P图形解码器6对从BD-ROM或HD读取的图形流进行解码,并且将得到的未压缩图形绘制到呈现图形平面7。随着图形流被解码和绘制,在屏幕上出现字幕。

[0297] 呈现图形平面7是具有用于存储一个屏幕的未压缩图形的区域的存储器。该平面的分辨率为 1920×1080 ,并且按照用索引色彩表示的8位像素数据的方式将未压缩图形存储在呈现图形平面7上。通过使用CLUT(色彩查找表)对索引色彩进行转换,可以呈现存储在呈现图形平面7上的未压缩图形。

[0298] 加法器8将未压缩画面数据(i)与存储在呈现图形平面7上的内容叠加在一起。

[0299] I图形解码器9对从BD-ROM或HD读取的IG流进行解码,并且将得到的未压缩图形绘制到互动图形平面10。

[0300] 经过I图形解码器9解码的未压缩图形数据被绘制到互动图形平面10。

[0301] 加法器11将互动图形平面10上的数据与加法器8的输出(即通过将未压缩画面数据和存储在呈现图形平面7上的内容叠加到一起从而得到的过渡视频数据)叠加到一起,从而得到最终的视频输出。

[0302] 音频解码器13对从De-MUX 3接收到的PES包进行解码从而输出未压缩音频数据。

[0303] CLUT单元14将存储在呈现图形平面7上的未压缩图形的索引色彩转换为Y、Cr和Cb值。

[0304] CLUT单元15将存储在互动图形平面10上的未压缩图形的索引色彩转换为Y、Cr和Cb值。

[0305] PSR集合16是再现装置内的寄存器集合。该寄存器集合包含64个播放器状态寄存器(PSR)和4096个通用寄存器(GPR)。这64个播放器状态寄存器代表了再现装置的当前状态。具体而言,PSR 5到PSR8代表了当前再现点。PRS 8的取值范围从0到0xFFFFFFFF,代表了用45KHz表示的当前再现点(当前PTM:呈现时间)。PSR 11代表了当前有效页(当前页)的page_id。PSR 10代表了当前页上处于选中状态的按钮(当前按钮)。

[0306] 系统时钟17生成时钟信号。De-MUX 3按照与时钟信号同步的方式执行PES包的转换。

[0307] user_timeout定时器18设置为在PTS($DS_n[ICS]$)所指定的时间的user_time_out_duration字段的值。当设置了user_time_out_duration字段值时,定时器18开始按照系统时钟进行倒计时,直到为零。每次接收到用户操作时,就将定时器18重新设置为user_

time_out_duration 字段值。如果在定时器 18 达到零之前未接收到任何用户操作,那么定时器 18 就发生超时。

[0308] selection_timeout 定时器 19 设置为在 PTS(DS_n[ICS]) 所指定的时间的 selection_time_out_pts 字段的值。当设置了 selection_time_out_pts 字段值时,定时器 19 开始按照系统时钟进行倒计时,直到为零。每次接收到用户操作时,就将定时器 19 重新设置为 selection_time_out_pts 字段值。如果在定时器 19 达到零之前未接收到任何用户操作,那么定时器 19 就发生超时。如果定时器 19 发生超时,就可以知道图 9 中所示的有效互动周期已经结束。

[0309] composition_timeout 定时器 20 设置为在 DTS(DS_n[ICS]) 所指定的时间 composition_time_out_pts 字段的值。当设置了 composition_time_out_pts 字段值时,定时器 20 开始按照系统时钟进行倒计时,直到为零。即使接收到用户操作,定时器 20 也不会重新设置。定时器 20 到达零,也就是定时器 20 发生超时的情况下,就知道已经到达了时元结束。

[0310] 如果 DS_n 是时元开始 DS,那么 effect_duration 定时器 21 设置为在 DTS(DS_n[ICS]) 所指定的时间 effect_duration 字段的值。当设置了 effect_duration 字段值时,定时器 21 开始按照系统时钟进行倒计时,直到为零。即使接收到用户操作,定时器 21 也不会重新设置。定时器 21 到达零,也就是定时器 21 发生超时的情况下,就知道需要显示下一个动画效果的显示成分。

[0311] 操作接收单元 22 接收遥控器或再现装置的前面板所发出的用户操作,并且将代表了接收到的用户操作的信息(此信息以后称作“U0”)输出到 I 图形解码器 9。

[0312] 状态控制单元 23 根据来自 I 图形解码器 9 的指令将希望得到的值设置到 PSR。可以通过以下方式给出这种指令:(1) 直接寻址或者(2) 间接寻址。如果是直接寻址,那么 I 图形解码器 9 就把将要设置的直接值输出到 PSR 集合 16 的适当的寄存器。如果是间接寻址, I 图形解码器 9 就输出关于再现装置或用户偏好的变化的通知。在接到这样一种通知时,状态控制器 23 就确定将要设置的值以便反映该变化,并且将通过这种方式确定的值设置到 PSR 集合 16 的适当的寄存器。状态控制器 23 通过执行“再现条件改变时执行的过程”或“要求改变时执行的过程”从而确定该值。接下来将描述设置 PSR 11(当前页)和 PSR10(当前按钮)所进行的过程。

[0313] 图 38A 是对 PSR 11 执行的“再现条件改变时执行的过程”的流程图。该过程用于将 ICS 内的第一个 page_info 结构设置到 PSR 11(步骤 S99)。

[0314] 图 38B 是对 PSR 11 执行的“要求改变时执行的过程”的流程图。假设接收到用户操作并且要求呈现具有页编号 X 的页。响应于该用户操作,状态控制器 23 判断 X 是否是有效值(步骤 S100)。如果 X 是有效值,那么 PSR 11 就被设置为 X(步骤 S101)。如果 X 是无效值,那么 PSR 11 就保持不变(步骤 S102)。

[0315] 按照上述方法改变 PSR 11 的值。接下来将描述关于 PSR 10 的“再现条件改变时执行的过程”或“要求改变时执行的过程”。

[0316] 图 39A 是对 PSR 10 执行的“再现条件改变时执行的过程”的流程图。

[0317] 首先,状态控制器 23 判断相关于当前页的 default_selected_button_id_ref 字段值是否有效(步骤 S111)。如果步骤 S111 中的判断结果是“是”,那么就将 default_

selected_button_id_ref 字段值设置到 PSR 10(步骤 S112)。

[0318] 如果 default_selected_button_id_ref 字段值是无效的,那么随后就判断 PSR 10 是否具有有效值(步骤 S113)。如果 PSR 10 是有效的,那么 PSR 10 就保持不变,以便保持当前具有的值(步骤 S114)。另一方面,如果 PSR 10 是无效的,那么就将 PSR 10 设置为用于标识当前页的第一 button_info 结构的 button_info 字段值(步骤 S115)。

[0319] 图 39B 是对 PSR 10 执行的“要求改变时执行的过程”的流程图。假设进行了用户操作,并且要求选择具有按钮编号 X 的按钮。响应于该用户操作,状态控制器 23 判断 X 是否是有效 button_id 字段值(步骤 S116)。如果 X 是有效值,那么 PSR 10 就被设置为值 X(步骤 S118)。如果 X 是无效值,那么 PSR 10 就保持不变并且因此保持了当前具有的值(步骤 S117)。通过上述过程,可以保证 PSR 10 和 11 总是保持当前值。上述是对再现装置的内部结构的描述。

[0320] <I 图形解码器 9 的内部结构 >

[0321] 接下来将参考图 40 描述 I 图形解码器 9 的内部结构。如图中所示,I 图形解码器包含编码数据缓存器 33、流图形处理器 34、对象缓存器 35、成分缓存器 36 以及图形控制器 37。

[0322] 编码数据缓存器 33 是一种缓存器,用于暂时存储 ICS、PDS 和 ODS 以及它们的 DTS 和 PTS。

[0323] 流图形处理器 34 对 ODS 进行解码并且将得到的未压缩图形对象输出到对象缓存器 35。

[0324] 对象缓存器 35 是一种缓存器,用于存储多个未压缩图形对象(用图中的方框代表),其中流图形处理器 34 对这些解压缩图形对象进行解码。

[0325] 成分缓存器 36 是一种存储器,用于存储 ICS。成分缓存器 36 向图形控制器 37 提供 ICS 内的 page_info 结构和每个 page_info 结构中的 button_info 结构。

[0326] 图形控制器 37 对存储在成分缓存器 37 中的 ICS 进行解码,然后根据解码后的 ICS 组合显示成分。也就是说,图形控制器 37 参考由 PSR 11 的值所标识的 page_info 结构(当前 page_info 结构)从对象缓存器 15 中读取图形对象,并且将读取的图形对象绘制到互动显示平面 10。这里所读取的图形对象是由相关于当前页的每个 button_info 结构的 normal_start_object_id_ref 字段值到 normal_end_object_id_ref 字段值所标识的一系列图形对象。对于 PSR 10 指定的 button_info 结构,由从对象缓存器 15 中读取的 selected_start_object_id_ref 字段值到 selected_end_object_id_ref 字段值标识一系列图形对象。在图 40 中,箭头 bg1、bg2、bg3 和 bg4 示意性地表示了由图形控制器 37 执行的图形对象绘制。通过这样的绘制,将包含有按钮 0-A、0-B、0-C 和 0-D 的页存储到互动图形平面 10 上并且与视频叠加到一起。

[0327] 上面是对图形控制器 37 执行的过程的大致描述。除此之外,图形控制器 37 还可以响应于 PSR 10 或 11 的值的变化的变化、定时器 18-21 的超时或者是成分超时定时器 20 接收到的用户操作来更新显示成分。图 41-47 详细地显示了图形控制器 37 所执行的过程。

[0328] 图 41 是图形控制器 37 所执行的过程的主程序的流程图。在该主程序中,步骤 S1-S6 形成了一个环路。在该环路的每次重复过程中,都要判断是否已经发生了特定的事件(步骤 S1-S4),多页菜单的一种显示成分被更新为随后一种显示成分,以便实现动画呈

现（步骤 S5），以及根据用户操作进行处理（U0 处理）（步骤 S6）。如果步骤 S1-S4 中的任意一个判断的结果为“是”，那么就执行相应的步骤，并且处理返回到主程序。

[0329] 在步骤 S1，判断当前再现点是否到达了由 PTS(DS_n[ICS]) 指定的点。如果步骤 S1 中的判断结果是“是”，那么就执行上述的“再现条件改变时执行的过程”。因此，就将 PAGE[0]，也就是多页菜单的第一页设置为当前页（步骤 S7）。然后，user_timeout 定时器 18、selection_timeout 定时器 19 以及 composition_timeout 定时器 20 都重新设置（步骤 S8），并且在屏幕上呈现当前页（步骤 S9）。然后，处理返回到步骤 S1，以便进行下一次环路重复。

[0330] 在步骤 S2，判断 user_timeout 定时器 18 是否已经超时。如果步骤 S2 的判断结果是“是”，那么就执行步骤 S11-14 的环路。在该环路的每次重复中，从屏幕上移除多页菜单的任意 PAGE[j]（步骤 S13），然后再现 PAGE[j] 的退出效果动画（步骤 S14）。重复地将该环路用于 ICS 中规定的 PAGE[0] 以及其后的页（步骤 S11 和 S12）。

[0331] 在步骤 S3，判断 selection_timeout 定时器 19 是否已经超时。如果步骤 S3 的判断结果是“是”，那么就自动激活当前页上的按钮（自动激活：步骤 S10）。然后，处理返回到步骤 S1，以便进行下一次环路重复。

[0332] 在步骤 S4，判断 composition_timeout 定时器 20 是否已经超时。如果步骤 S4 的判断结果是“是”，那么从屏幕上移除 PAGE[0]，也就是多页菜单的第一页（步骤 S15），然后再现 PAGE[0] 的退出效果动画（步骤 S16）。然后，处理返回到步骤 S1，以便进行下一次环路重复。

[0333] 接下来将描述多页菜单的首次呈现。在状态控制信息 23 执行上述的“再现条件改变时执行的过程”从而将 PAGE[0] 设置为当前页之后，通过图 42 中所示的步骤 S15-S23 执行该首次呈现。在步骤 S24，执行当前页的引入效果动画。在随后的步骤 S25 中，执行当前按钮设置过程。由 PSR 10 指定该当前按钮。因此，在步骤 S25，对 PSR 10 执行“再现条件改变时执行的过程”，以便确定当前按钮。然后，处理转移到步骤 S17-S22。

[0334] 步骤 S17 和 S22 形成了一个环路，并且对当前页中每个 button_info 结构都重复该环路（步骤 S17 和 S18）。将环路的当前重复过程中所处理的 button_info 结构指定为 button_info(p)。

[0335] 在步骤 S19，判断 button_info(p) 是否相关于当前按钮。如果步骤 S19 中的判断结果是“是”，接下来就执行步骤 S20。如果不是如此，接下来就执行步骤 S21。

[0336] 在步骤 S21，将对象缓存器 35 的图形对象中由 normal_start_object_id_ref 字段值所指定的图形对象指定为图形对象 (p)。

[0337] 在步骤 S20，将对象缓存器 35 的图形对象中由 selected_start_object_id_ref 字段值所指定的图形对象指定为图形对象 (p)。

[0338] 一旦在步骤 S20 或 S21 中指定了图形对象 (p)，就将该图形对象 (p) 绘制到互动图形平面 10 上的由 button_horizontal_position 和 button_vertical_position 字段值所指定的位置处（步骤 S22）。通过对当前页中每个 button_info 结构都重复执行上述步骤，在相关于每种按钮状态的多个图形对象中，第一个图形对象被绘制到互动图形平面 10 以使用于每个按钮。在重复进行上述步骤之后，对 CLUT 单元 15 进行设置，以便使用由当前页的 pallet_id_ref 字段值指定的调色板数据呈现当前页（步骤 S23）。上面就是对图 42 中

所示的流程图的描述。

[0339] 图 43 是再现引入效果动画的过程的流程图。在该流程图中,变量 t 表示用于定义引入效果的 effect_sequence 中的显示成分中的任意一个显示成分。另外,变量 u 表示用于 effect(t) 的 composition_object 中任意一个 composition_object。如该流程图所示,首先,将变量 t 和 u 初始化为“0”(步骤 S30)。然后,effect_duration 定时器 21 设置为 effect(t).effect_duration 字段的值(步骤 S31),并且对 CLUT 单元 14 和 15 进行设置,从而使用由 effect(t).pallet_id_ref 字段值指定的调色板数据进行呈现(步骤 S32)。然后,执行步骤 S33-S40 的环路。该环路为两级环路。第一级环路(步骤 S33-S37)将变量 u 用作控制变量,而第二级环路(步骤 S33-S40)将变量 t 用作控制变量。

[0340] 在第一级环路中,判断相关于 effect(t) 的 composition_object(u) 结构内提供的 object_cropped_flag 字段是否设置为“0”(步骤 S33)。如果该字段设置为“0”,那么对于 composition_object(u) 来说并不在屏幕上呈现图形对象(步骤 S35)。另一方面,如果该字段设置为“1”,那么根据 composition_object(u) 的 object_cropping_horizontal_position、object_cropping_vertical_position、cropping_width 以及 cropping_height 字段从而裁切对象。然后将图形对象裁切得到的部分绘制到由 composition_object(u) 的 window_id_ref 字段指定的窗口内,而在窗口内的位置由 composition_object(u) 的 composition_object_horizontal_position、composition_object_vertical_position 字段指定(步骤 S34)。然后,变量 u 增加“1”。第一级环路一直重复进行,直到变量 u 等于 number_of_composition_object 字段值。

[0341] 对于第一级环路的每次重复(步骤 S33-S37),在第二级环路中将执行下述步骤。在 effect_duration 超时后(步骤 S38),变量 t 增加“1”,并且变量 u 初始化为“0”(步骤 S39)。第二级环路一直重复进行,直到变量 t 等于 number_of_effect(t) 字段值(步骤 S40)。上面就是对再现引入效果的过程的描述。注意到并未描述再现退出效果的过程,这是因为该过程基本上与上述过程相同。

[0342] 在再现完引入效果之后,立即呈现当前页并且更新首次显示以便进行动画呈现。通过将相关于每个 button_info 结构中提供的 normal_start_object_id_ref 和 selected_start_object_id_ref 字段值的图形对象绘制到互动图形平面 10 从而呈现当前页的第一显示成分。通过在主程序的环路的每次重复过程中更新互动图形平面 10 上的按钮图像从而按照动画方式呈现每个按钮。通过用相关于每个按钮的一系列图形对象中任意一个(第 q 个图形对象)重写互动图形平面 10 从而更新每个按钮图像。也就是说,通过将相关于每个 button_info 结构内提供的 normal_state_info 和 selected_state_info 字段的一系列图形对象逐个绘制到互动图形平面 10 上从而以动画方式呈现每个按钮。注意到变量 q 用于指定相关于每个 button_info 结构内提供的 normal_state_info 和 selected_state_info 字段的单个图形对象。

[0343] 接下来将参考图 44 描述用于实现以动画方式呈现的显示更新的过程。

[0344] 在步骤 S41,判断是否已经呈现了第一显示成分。如果还未呈现第一显示成分,那么处理就返回到主程序,而不执行任何操作。另一方面,如果已经呈现了第一显示成分,就执行步骤 S42-S53。步骤 S41-S55 形成了一个环路,并且在该环路中对于 ICS 中的每个 button_info 结构都要重复进行步骤 S44-S55(步骤 S42 和 S43)。

[0345] 在步骤 S44, 变量 q 设置为对应于 button_info(p) 结构的变量 animation(p) 的值。因此, 变量 q 代表 button_info(p) 的当前帧序号。

[0346] 在步骤 S45, 判断 button_info(p) 是否对应于当前正处于选中状态的按钮 (即当前按钮)。

[0347] 如果 button_info(p) 并不对应于当前按钮, 那么就将变量 q 与 button_info(p) 的 normal_start_object_id_ref 相加从而获得 ID(q) (步骤 S46)。

[0348] 如果 button_info(p) 对应于当前按钮, 那么就执行步骤 S47。

[0349] 在步骤 S47, 判断当前按钮是否处于激活状态。如果在步骤 S47 中的判断结果是“是”, 那么就将变量 q 与 button_info(p) 的 activated_start_object_id_ref 字段值相加从而获得 ID(q) (步骤 S54)。然后, 执行相关于 button_info(p) 的一个导航命令 (步骤 S55)。

[0350] 另一方面, 如果当前按钮并不处于激活状态, 那么就将变量 q 与 button_info(p) 的 selected_start_object_id_ref 字段值相加从而获得 ID(q) (步骤 S48)。

[0351] 一旦获得了 ID(q), 就将存储在对象缓存器 35 的图形对象中 ID(q) 所标识的那个图形对象绘制到互动图形平面 10。在互动图形平面 10 上的绘制位置由 button_info(p) 中 button_horizontal_positon 和 button_vertical_positon 字段指定。

[0352] 通过环路重复, 就将相关于处于选中 (或者激活) 状态的当前按钮的第 q 个图形对象绘制到互动图形平面 10。

[0353] 在步骤 S50, 判断 normal_start_object_id_ref 字段值与变量 q 之和是否达到 normal_end_object_id_ref 字段值。如果该和未达到 normal_end_object_id_ref 字段值, 那么变量 q 就加“1”, 并且将增加后的变量 q 设置为变量 animation(p) (步骤 S51)。另一方面, 如果该和已经达到 normal_end_object_id_ref 字段值, 那么就判断 repeat_flag 字段是否设置为“1” (步骤 S52)。如果 repeat_flag 字段设置为“1”, 那么就将变量 animation(p) 初始化为“0” (步骤 S53)。对于 ICS 内的每个 button_info 结构都重复进行上述步骤 (步骤 S42 和步骤 S43), 并且随后处理返回到主程序。

[0354] 通过上述步骤 S41-S53, 每次执行主程序时都使用一个新的图形对象对呈现在屏幕上的每个按钮图像进行了更新。因此, 通过重复主程序, 可以按照动画的形式呈现按钮图像。当按照动画方式呈现按钮时, 图形控制器 37 调整用于呈现每个图形对象的持续时间, 以便保持 animation_frame_rate_code。上面就是对动画方式呈现的过程的描述。

[0355] 在上述的步骤 S47 中, 如果判断出当前按钮处于激活状态, 那么就需要在步骤 S55 执行相关于当前按钮的导航命令。图 45 是执行导航命令的过程的流程图。首先, 判断该导航命令是否为 SetButtonPageCommand (步骤 S56)。如果该导航命令不是 SetButtonPageCommand, 那么就只是执行该导航命令 (步骤 S57)。另一方面, 如果该导航命令是 SetButtonPageCommand, 那么就向状态控制单元 23 发出指令, 以便将该导航命令的运算符所指定的页序号指定为页序号 X, 以及将该导航命令的运算符指定的按钮序号设置到 PSR 10 (步骤 S58)。然后, 状态控制器 23 对 PSR 11 执行“要求改变时执行的过程” (步骤 S59)。如上面所述, PSR 11 具有代表了当前所呈现的页 (即当前页) 的值。因此, 通过对 PSR 11 执行“要求改变时执行的过程”, 就可以确定当前页。然后, 状态控制单元 23 对 PSR 10 执行“再现条件改变时执行的过程” (步骤 S60)。上面就是对图 45 中所示流程图的描

述。

[0356] 上面就是对执行导航命令的过程的描述。如上面所述,当激活相关的按钮时,就会执行该导航命令。响应于用户操作(U0),通过下述过程呈现这些按钮状态变化。图 46 显示了处理 U0 的过程。如该流程图中所示,在步骤 S61-S64 判断是否发生了特定的事件。每种事件的发生都要求执行特定的步骤。然后,处理返回到主程序。具体而言,在步骤 S61,判断 U0_mask_table 字段是否被设置为“1”。如果该字段被设置为“1”,那么处理返回到主程序,而不执行任何步骤。

[0357] 在步骤 S62,判断是否按下了遥控器上的向上、下、左和右移动键中的任意一个。当按下这些键中的任意一个时,user_timeout 定时器 18 和 selection_timeout 定时器 19 都被重新设置(步骤 S65)。然后,将另一个按钮指定为新的当前按钮(步骤 S66),然后判断新指定的当前按钮的 auto_action_flag 是否被设置为“01”(步骤 S67)。如果 auto_action_flag 未被设置为“01”,那么处理就返回到主程序。另一方面,如果 auto_action_flag 被设置为“01”,那么就激活当前按钮(i)(步骤 S69)。然后,将变量 animation(i) 设置为“0”(步骤 S70)。

[0358] 在步骤 S63,判断是否做出了数字输入。如果做出了数字输入,那么 user_timeout 定时器 18 和 selection_timeout 定时器 19 就重新设置(步骤 S71),并且将另一个按钮指定为新的当前按钮(步骤 S72)。然后处理返回到主程序。

[0359] 在步骤 S64,判断是否按下了遥控器上的激活按键。当按下激活按键时,user_timeout 定时器 18 和 selection_timeout 定时器 19 重新设置(步骤 S68),并且随后激活当前按钮(i)(步骤 S69)。在按钮状态转换之后,变量 animation(i) 被设置为“0”(步骤 S70)。注意到上述用于指定新的当前按钮的步骤 S66 是通过调用图 47 中所示的子程序从而实现的。接下来将参考该图描述该子程序。

[0360] 图 47 是当前按钮改变过程的流程图。首先,确定相关于当前按钮的 neighbor_info 中的 upper_button_id_ref、lower_button_id_ref、left_button_id_ref 以及 right_button_id_ref 字段中的哪个对应于按下的按键(步骤 S75)。

[0361] 这里,用 button(Y) 表示当前按钮,并且用按钮 X 表示 upper_button_id_ref、lower_button_id_ref、left_button_id_ref 以及 right_button_id_ref 字段中的一个指定的按钮(步骤 S76)。然后,状态控制单元 23 对 PSR 10 执行“要求改变时执行的过程”(步骤 S77)。因此,PSR 10 设置为值 X。

[0362] 此后,变量 animation(X) 和 animation(Y) 设置为“0”(步骤 S78),并且返回到主程序。

[0363] 如上面所述,在 U0 处理中,响应于 U0 从而激活按钮。但是,在 selection_time_out_pts 发生超时的情况下还要执行转换到激活状态的按钮状态转换。接下来将参考图 48,描述在 selection_time_out_pts 发生超时的情况下的按钮自动激活的过程。

[0364] 图 48 是按钮的自动激活过程的流程图。首先,判断 default_activated_button_id_ref 字段是否设置为“00”或者“FF”(步骤 S91)。如果该字段设置为“00”,那么处理就返回到主程序,而不执行任何步骤。另一方面,如果该 default_activated_button_id_ref 字段设置为“FF”,那么就激活当前按钮(i)(步骤 S93)。此后,对应于当前按钮(i)的变量 animation(i) 设置为“0”(步骤 S94),并且返回到主程序。

[0365] 如果 default_activated_button_id_ref 字段值既不是“00”也不是“FF”，那么就将 default_activated_button_id_ref 字段值所指定的按钮确定为新的当前按钮 (i) (步骤 S92)。因此，default_activated_button_id_ref 字段值所指定的按钮就接收了激活状态 (步骤 S93)。此后，对应于当前按钮 (i) 的变量 animation(i) 设置为“0” (步骤 S94)，并且返回到主程序。

[0366] 通过上述步骤，处于选中状态的按钮在预先确定的持续时间的结束时自动地设置为激活状态。上面就是对图 47 中所示的流程图的描述。

[0367] <弹出 U/I 中图形显示的打开 / 关闭>

[0368] 上面的描述涉及在 user_interface_model 显示为总是开启 U/I 的情况下，I 图形解码器 9 执行的过程。当使用弹出 U/I 时，I 图形解码器 9 按照下述方式进行工作。首先，与总是开启 U/I 情况中类似，I 图形解码器 9 执行解码从而获得对象缓存器 35 中解码后的图形对象。I 图形解码器 9 随后在 PTS(DS_n[ICS]) 所示的时间将图形对象绘制到互动图形平面 10。到这一点为止，I 图形解码器 9 的工作方式与总是开启 U/I 情况下的工作方式是类似的。但是，此后的所执行的处理不同，其中不同之处在于图形控制器 37 并不叠加存储在互动图形平面 10 上的页，因此也就不呈现页。也就是说，图形控制器 37 通过不将存储在互动图形平面 10 上的数据输出到 CLUT 单元 15 从而“关闭”该页。由于页并未呈现在屏幕上 (即关闭)，图形控制器 37 等待用户的 Pop-Up_on 操作。响应于用户的 Pop-Up_on 操作，图形控制器 37 将存储在互动图形平面 10 上的图形数据输出到 CLUT 单元 15，并且在该单元中图形数据与视频数据叠加在一起。因此，页被“打开”或呈现在屏幕上。如上面所述，图形控制器 37 打开和关闭图形显示从而实现弹出显示。

[0369] 在此后进行的操作中，总是开启 U/I 和弹出 U/I 这两种情况之间并不存在区别。当激活主页上的任意一个按钮时，都会执行关于该激活按钮的导航命令。如果 selection_timeout 定时器 19 发生超时，就会自动激活主页上的当前按钮，并且在屏幕上呈现子页。

[0370] 当用户操作为 Pop-Up_Off 时，图形控制器 37 关闭所有屏幕上的页，由此使得屏幕上不呈现页。

[0371] 如果在一定时间周期内无用户操作，那么 user_timeout 定时器 18 就倒计时到零。如果 user_timeout 定时器 18 发生超时，那么图形控制器 37 也会关闭所有屏幕上的页，由此使得屏幕上不呈现页。通过执行上述的打开和关闭过程从而实现图 12A 中所示的状态转换。

[0372] 如上面所述，通过打开和关闭操作，实现了弹出 U/I。

[0373] 如上面所述，本发明的第六实施例实现了可以支持具有第一到第五实施例中所述的数据结构的 BD-ROM 的再现装置的工业生产。这有助于根据第一到第五实施例的 BD-ROM 的推广使用。

[0374] (第七实施例)

[0375] 本发明的第七实施例涉及 BD-ROM 的制造过程。图 49 是根据第一到第五实施例的 BD-ROM 的制造过程。

[0376] 该制造过程包含记录视频、音频等等的材料生产步骤 (步骤 S201)；使用创作设备生成应用格式的创作步骤 (步骤 S202)；以及压制步骤，其中生成 BD-ROM 的原始主版 (master) 并且执行冲压和黏合从而完成 BD-ROM 的制造 (步骤 S203)。

[0377] 该 BD-ROM 创作步骤包含下述 S204-S213 的步骤。

[0378] 接下来将描述步骤 S204-S213。在步骤 S204,生成控制信息、调色板定义信息以及图形。在步骤 S205,将控制信息、调色板定义信息以及图形转换为功能部分。在步骤 S206,根据需要与其保持同步的画面的显示定时从而设置每个 ICS 的 PTS。在步骤 S207,根据 PTS[IDS] 设置 DTS[ODS] 和 PTS[ODS]。在步骤 S208,根据 DTS[ODS] 设置 DTS[ICS] 和 PTS[PDS]。

[0379] 在步骤 S209,用图形表示播放器模型中每个缓存器的占据情况的变化。在步骤 S210,判断用图形表示的变化是否满足播放器模型的限制。如果判断结果是否定的,那么就在步骤 S211 重写每个功能部分的 DTS 和 PTS。如果判断结果是肯定的,那么就在步骤 S212 生成图形流,并且在步骤 S213 将该图形流与分开生成的视频流和音频流复用到一起从而形成 AV 剪辑。此后,该 AV 剪辑适用于蓝光光盘只读格式,从而完成应用格式。

[0380] (变化形式)

[0381] 尽管已经通过上述实施例对本发明进行了描述,但是本发明并不限于这些具体的实施例。可以通过下述的变化形式 (A) 到 (L) 中的任意一个实现本发明。在本申请的每项权利要求中叙述的本发明包含上述实施例和下述它们的变化形式的扩展和概括。扩展和概括的程度依赖于做出本发明时本发明技术领域的现有技术。

[0382] (A) 上述实施例描述了将 BD-ROM 用作记录介质的情况。但是,本发明的主要特征在于记录在记录介质上的图形流,而这种图形流又不依赖于 BD-ROM 的物理特性。因此,本发明可以应用于能够记录图形流的任意记录介质。这样的记录介质的例子包含:光盘,例如 DVD-ROM、DVD-RAM、DVD-RW、DVD-R、DVD+RW、DVD+R、CD-R 或者 CD-RW;磁光盘,例如 PD 或 MO;半导体存储卡,例如 CompactFlash 卡、SmartMedia 卡、Memory Stick 卡、MultiMedia 卡或者 PCMCIA 卡;磁盘,例如软盘、SuperDisk、Zip 或者 Click!;可移动硬盘驱动器,例如 ORB、Jaz、SparQ、SyJet、EZFley 或者 Micordrive 以及不可移动硬盘驱动器。

[0383] (B) 上述实施例描述了其中再现装置对 BD-ROM 上的 AV 剪辑进行解码并且将解码后的 AV 剪辑输出到电视的情况。可替换的,该再现装置可以仅配有 BD 驱动器,而其它的结构元件位于电视中。在这种情况下,可以通过 IEEE 1394 连接器将该再现装置和电视包含在家庭网络中。上述实施例描述了其中再现装置连接到电视的情况,但是也可以将再现装置与显示设备集成到一起。另外,再现装置可以仅包含用于执行必要部分处理的系统 LSI(集成电路)。该再现装置和集成电路都是本说明书所描述的发明。因此,无论是对于该再现装置还是对于该集成电路,根据第六实施例中所述再现装置的内部结构所进行的再现装置制造过程都是实施本发明的过程。另外,对该再现装置的收费转让(即销售)或免费转让(即,礼品)、租赁以及进口都是实施本发明。同样地,通过店前橱窗展示、产品目录或宣传册等形式从而提供对该再现装置的转让或租赁也是实施本发明。

[0384] (C) 使用流程图中所示的程序所进行的信息处理实际上是通过硬件资源实现的。因此,描述了流程图中所示的过程步骤的程序其本身也是一种发明。上述实施例都涉及其中程序包含在再现装置中的情况,但是也可以独立于再现装置而单独使用这些程序。对这些程序的实施包含(1)制造;(2)收费转让或免费转让;(3)租赁;(4)进口;(5)通过双向电子通信网络向公众提供以及(6)使用店前橱窗展示、产品目录或宣传册等形式从而提供转让或租赁。

[0385] (D) 每个流程图中按照一定的时间顺序执行的步骤的时间元素可以被视作本发明的必要元素。因此,通过这些流程图所示的再现方法是一种发明。如果通过按照一定的时间顺序执行这些步骤从而执行每个流程图所示的处理,以便实现预定的目的和效果,那么就将其视作是实施本发明的再现方法。

[0386] (E) 在将 AV 剪辑记录到 BD-ROM 上时,可以向 AV 剪辑的每个 TS 包中加入一个扩展报头。该扩展报头称作 TP_extra_header,它包含 arrival_time_stamp 和 copy_permission_indicator,并且它的数据长度为 4 字节。具有 TP_extra_header 的 TS 包(此后称作“EX TS 包”)以 32 个包为单位进行分组,并且每组被写入到三个扇区。包含有 32 个 EX TS 包的一个组具有 6144 个字节(= 32×192),等效于三个扇区的大小,也就是 6144 个字节(= 2048×3)。包含在三个扇区中的这 32 个 EX TS 包称作“对准单元”。

[0387] 在通过 IEEE 1394 连接器连接的家庭网络中,再现装置按照下述方式传输对准单元。再现装置移除对准单元中 32 个 EX TS 包的每个中的 TP_extra_header,根据 DTCP 规范对每个 TS 包的主体进行加密,然后输出加密后的 TS 包。当输出 TS 包时,再现装置在相邻的 TS 包之间插入同步包。根据 TP_extra_header 的 arrival_time_stamp 所示的时间从而确定插入同步包的位置。再现装置输出 DTCP_descriptor 以及 TS 包。DTCP_descriptor 对应于 TP_extra_header 中的 copy_permission_indicator。通过提供指示了“禁止复制”的 DTCP_descriptor,在通过 IEEE 1394 连接器所连接的家庭网络中使用 TS 包时,可以防止 TS 包被记录到其它设备。

[0388] (F) 上述实施例涉及其中具有蓝光光盘只读格式的 AV 剪辑用作数字流的情况,但是还可以使用具有 DVD-Video 格式或 DVD-Video 记录格式的 VOB(视频对象)实现本发明。VOB 是一种符合 ISO/IEC13818-1 标准的程序流,并且通过将视频流和音频流复用到一起从而获得 VOB。另外,AV 剪辑中的视频流可以是 MPEG4 视频流或 WMV 视频流。而且,AV 剪辑中的音频流可以是线性 PCM 音频流、DolbyAC-3 音频流、MP3 音频流、MPEG-AAC 音频流或 dts 音频流。

[0389] (G) 可以通过对模拟广播所发出的模拟图像信号进行编码从而获得上述实施例中所述的电影。另外,该电影也可以是由数字广播所发出的传输流所组成的流数据。

[0390] 可替换地,可以通过对记录在视频带上的模拟/数字图像信号进行编码从而获得内容。另外,可以对视频照相机所直接捕获的模拟/数字图像信号进行编码从而获得内容。还可以应用通过发行服务器分发的数字作品。

[0391] (H) 上述实施例中描述的图形对象是行程长度编码光栅数据。行程长度编码用于图形对象的压缩/编码,这是因为行程长度编码适用于字幕的压缩和解压缩。字幕具有这样一种特性:水平方向上相同像素值的连续长度相对较长。因此,通过使用行程长度编码实现的压缩,可以获得高的压缩比。另外,行程长度编码可以降低解压缩时的负担,并且因此适用于通过软件实现解码。但是,对于本发明来说,并不一定需要对图形对象应用行程长度编码。例如,图形对象可以是 PNG 数据。另外,图形对象可以是矢量数据而不是光栅数据。而且,图形对象可以是透明图案。

[0392] (I) 可以确定传送速率 Rc 以便在垂直消隐时间内完成图形平面的清除和绘制。假设垂直消隐时间是 1/29.93 秒的 25%。那么 Rc 就被确定为 1Gbps。通过按照这种方式确定 Rc,可以平滑地显示图形,而这对于实现实际应用而言具有重大意义。

[0393] 另外,可以同时使用同步于线扫描的写入和垂直消隐时间内的写入。这可以确保在传送速率 R_c 为 256Mbps 的情况下也可以实现平滑呈现。

[0394] (J) 上述实施例涉及其中再现装置配有图形平面的情况。可替换的,再现装置可以包含用于存储一行未压缩像素的线缓存器。因为对于每个水平行(线)来说都要进行到画面信号的转换,所以可以使用线缓存器等效地进行到图像信号的转换。

[0395] (K) 如果出现多再现路径的合并点,那么需要根据已经被采用的再现路径从而选择另一个按钮作为缺省按钮。因此,在静态情况下希望规定这样一种再现控制,以便在实际选取再现路径时将对于每个再现路径来说是唯一的值存储到再现装置的寄存器中。可以设置再现过程从而使寄存器值所指定的按钮成为选中状态。通过这种方式,根据已经采用的再现路径可以使另一个按钮成为选中状态。

[0396] (L) 第六实施例中描述的图形平面优选地具有一种双缓存器结构。如果图形平面具有双缓存器结构,那么即使需要绘制大尺寸图形数据时(需要对应于多个帧的时间),也可以通过在两个缓存器之间进行切换从而即时地呈现每个显示成分。因此,双缓存器结构可以有效地应用于需要呈现具有全屏尺寸的菜单的情况。

[0397] 工业应用

[0398] 根据本发明的再现装置适用于个人使用,例如用于家庭影院系统。因为本发明的上述实施例公开了内部结构,所以可以大规模地生产具有这种内部结构的再现装置。因此,根据本发明的再现装置就其质量而言是工业上可使用的,并且具有工业应用性。

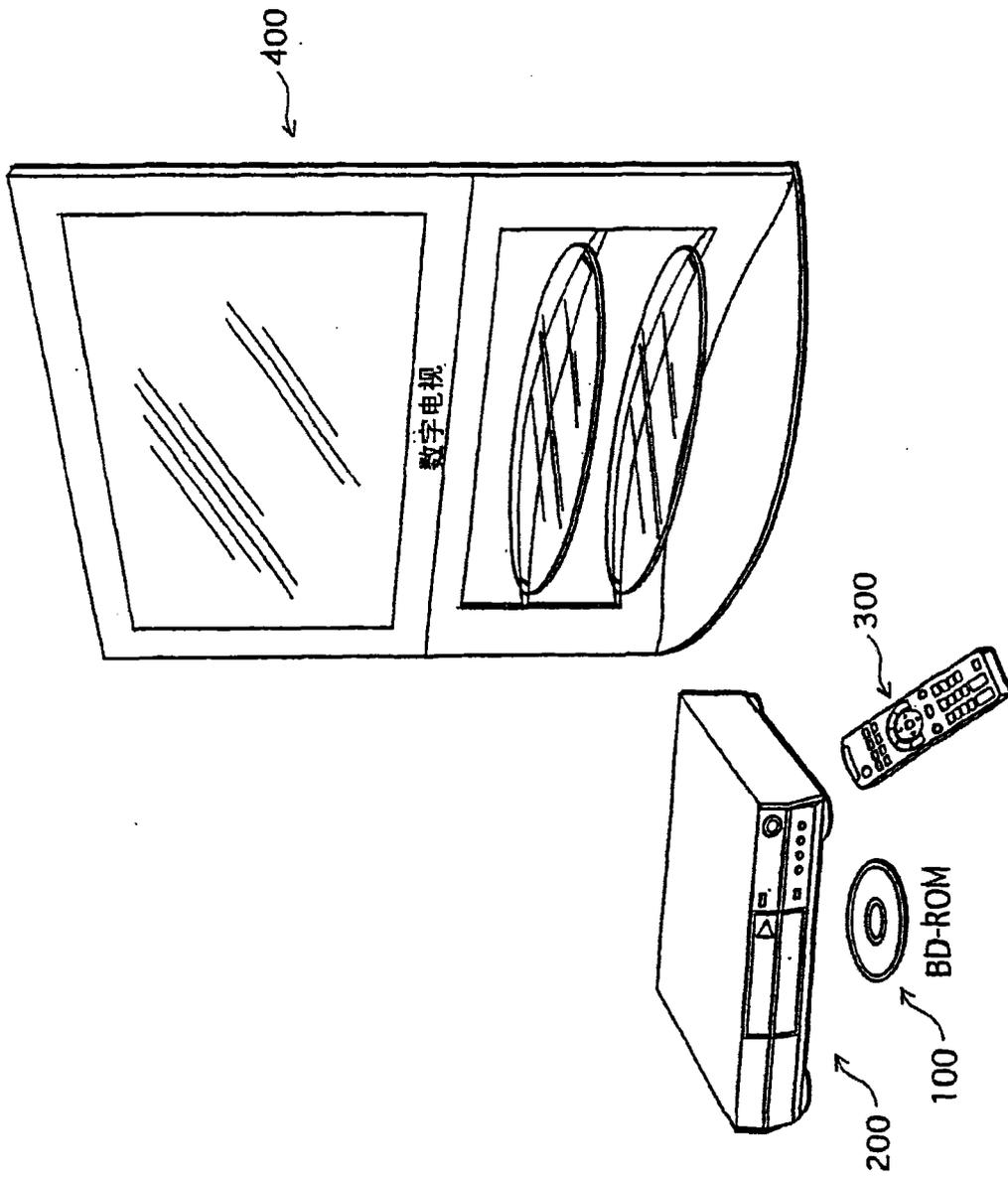


图 1

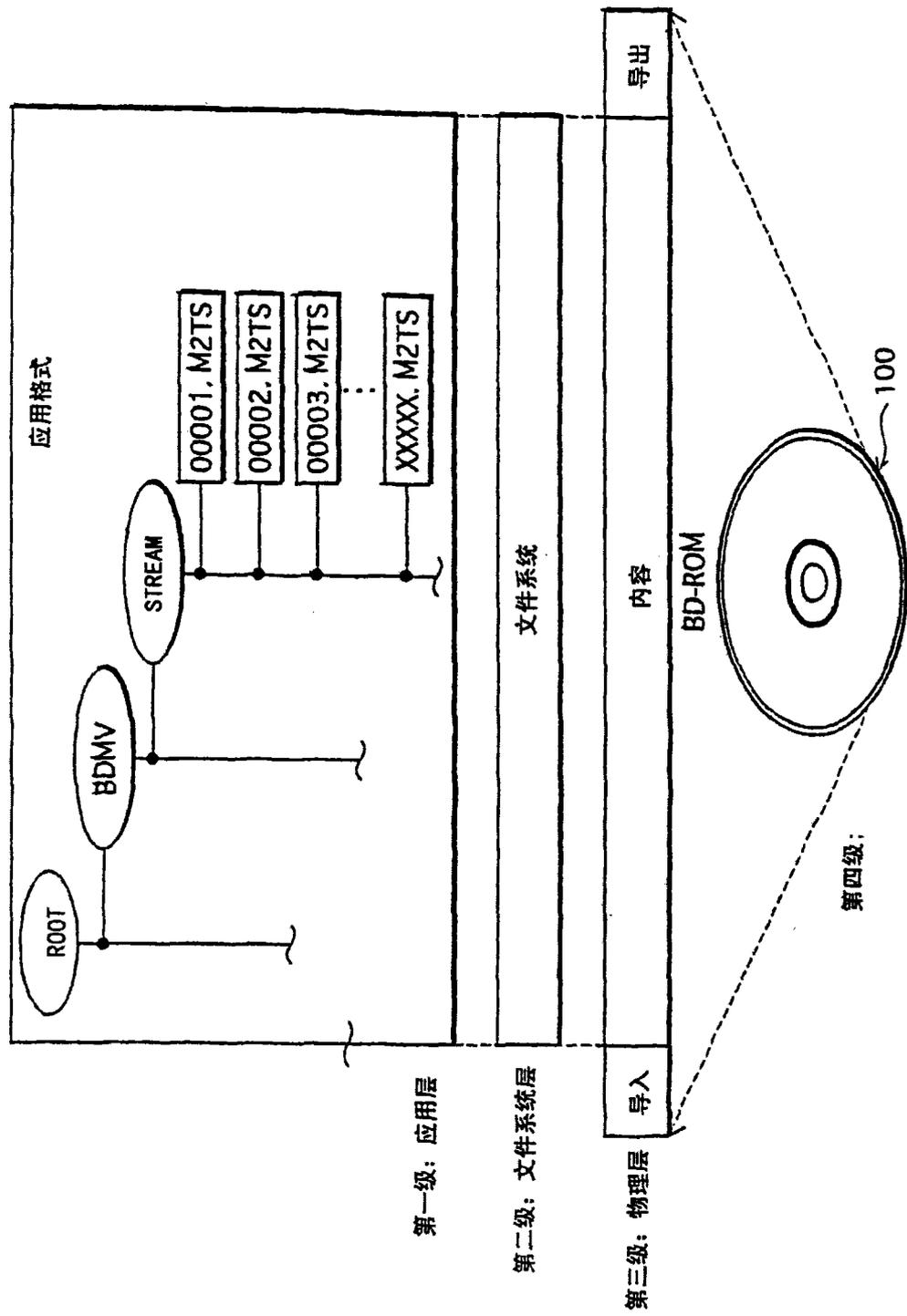


图 2

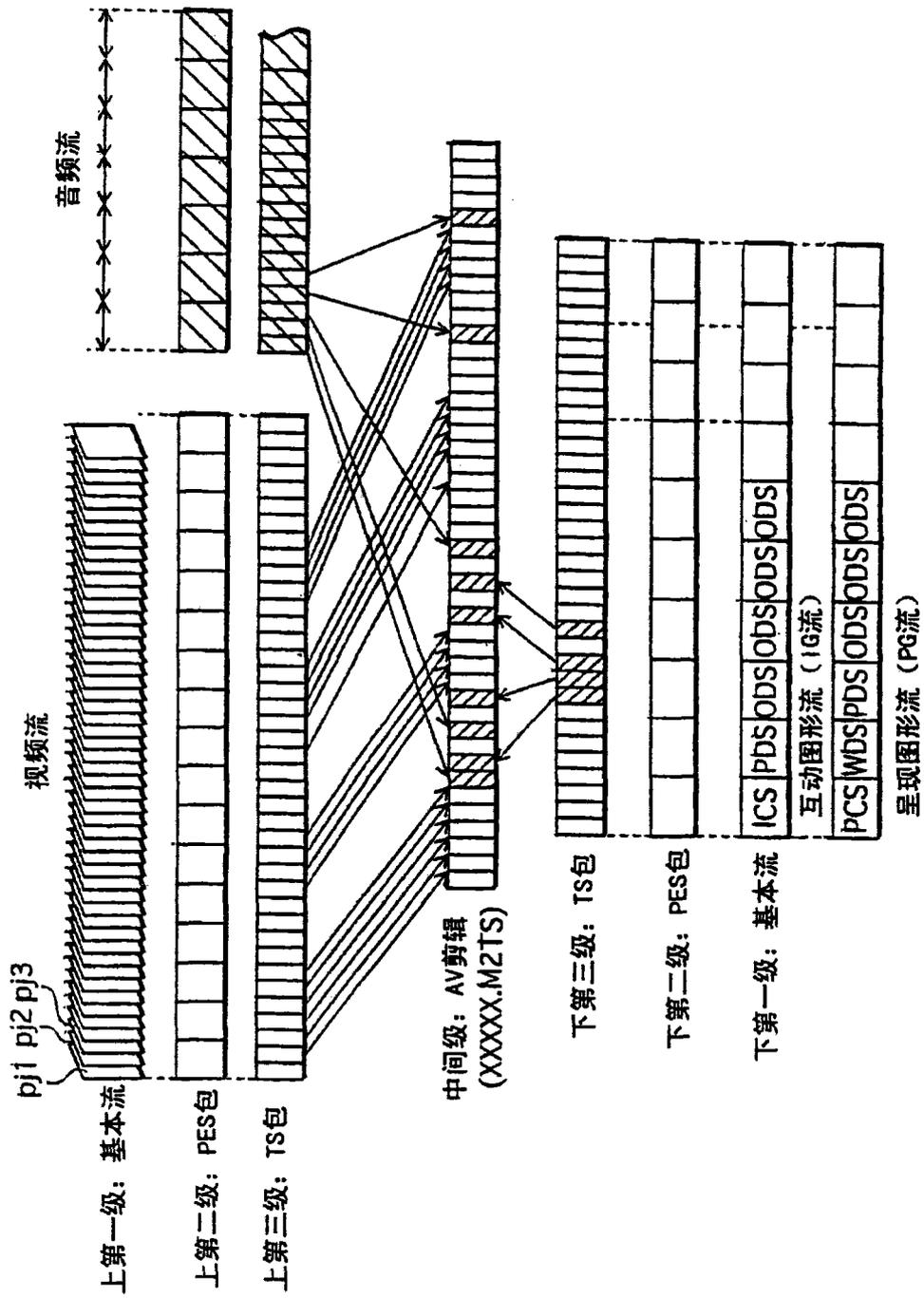


图 3

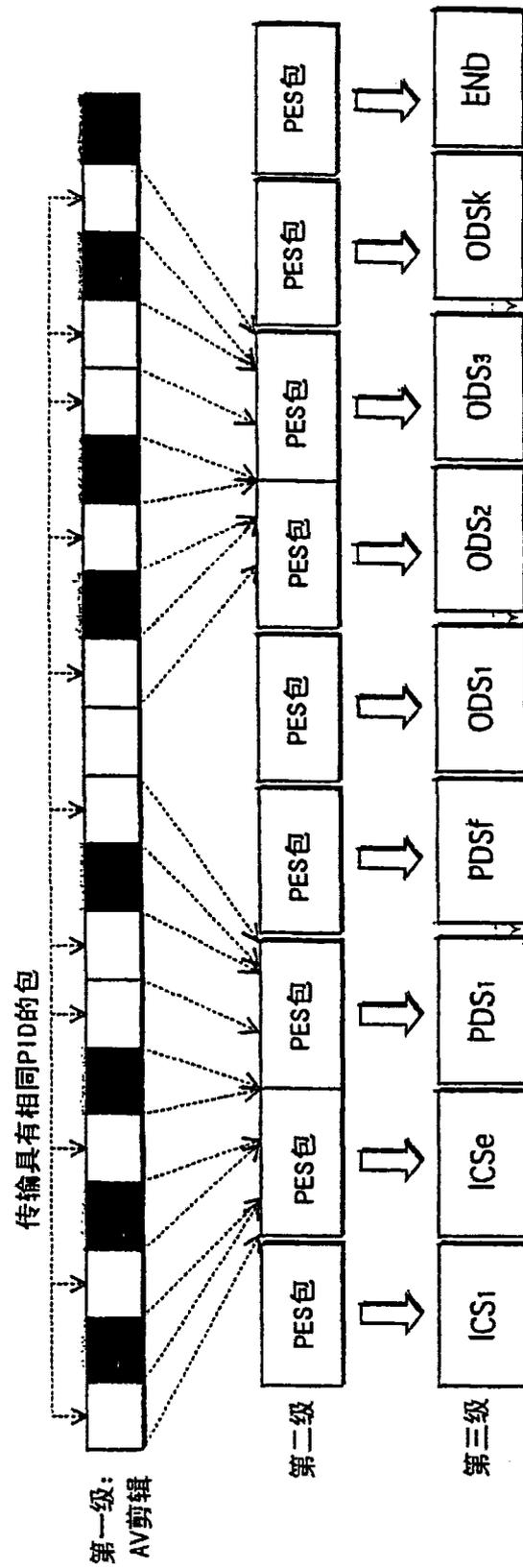


图 4A

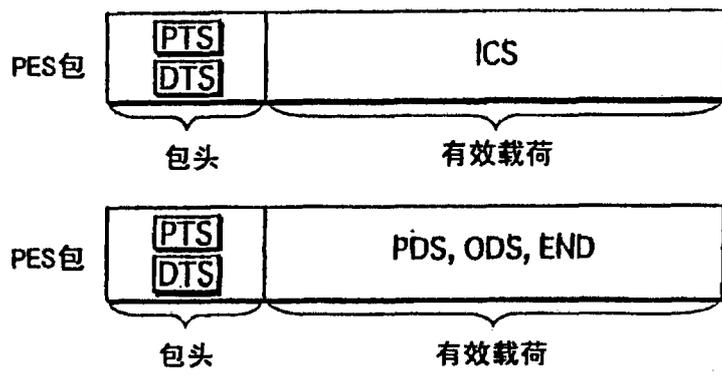


图 4B

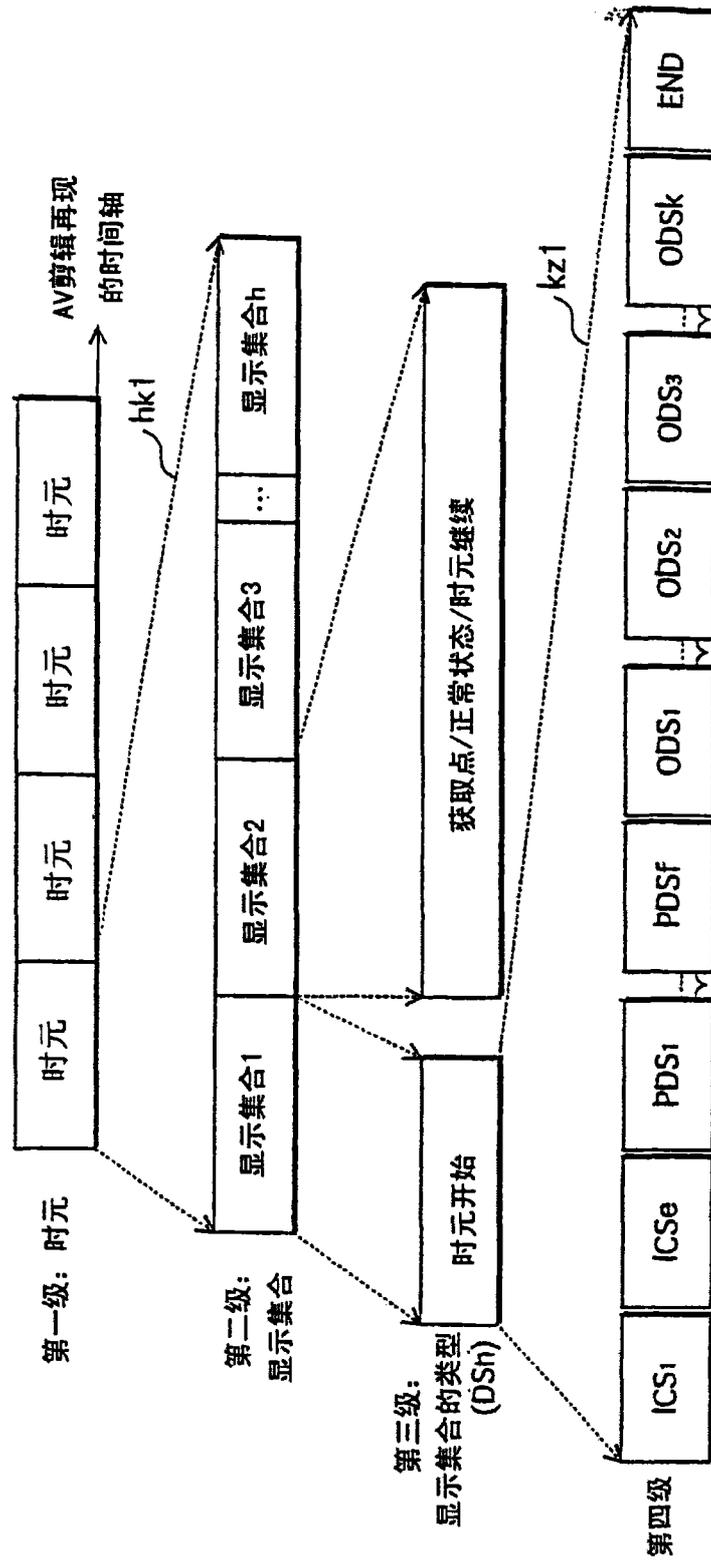


图 5

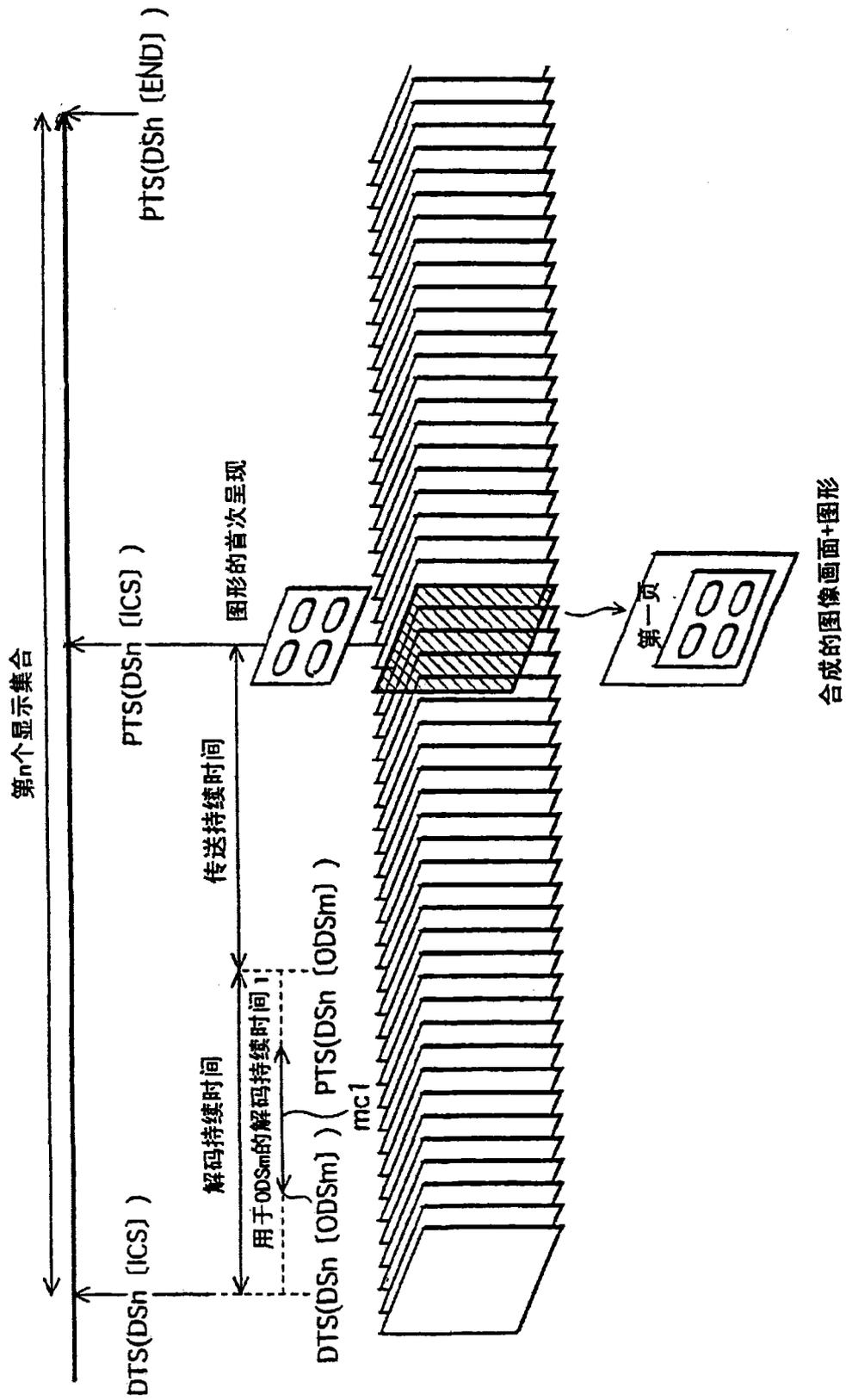


图 6

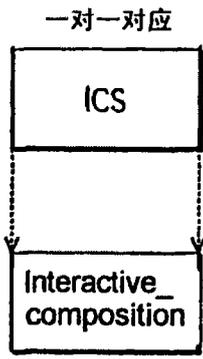


图 7A

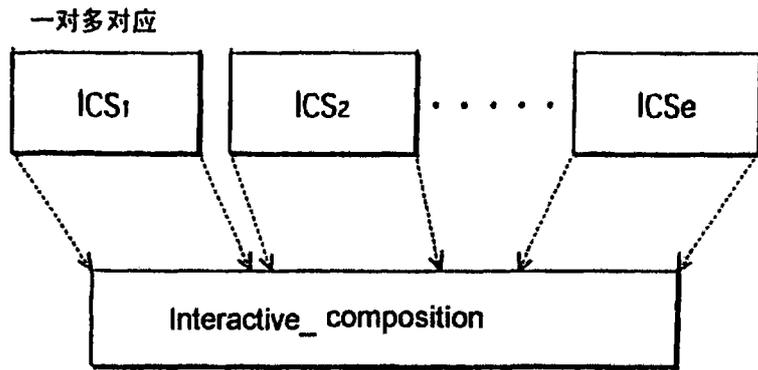


图 7B

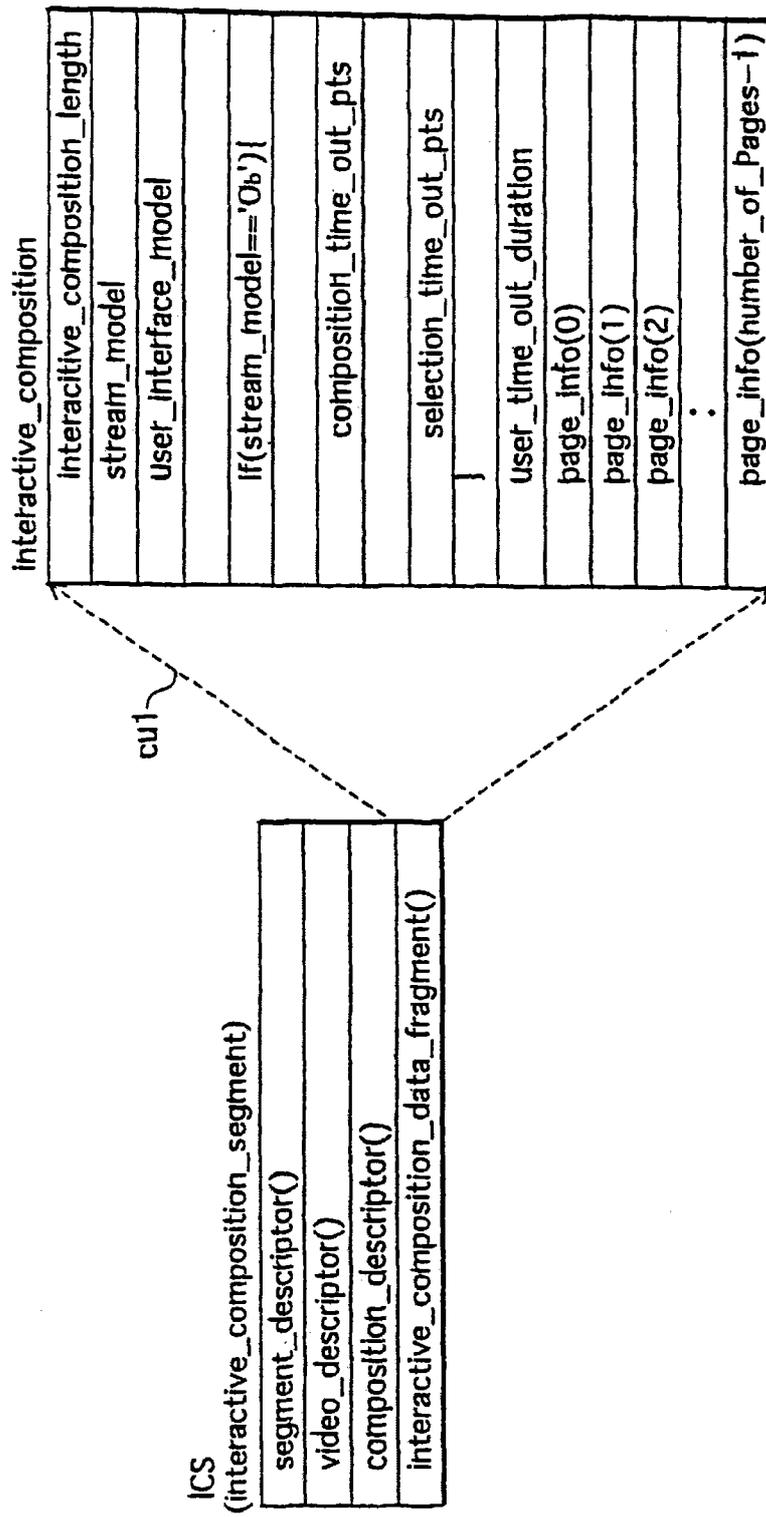


图 8

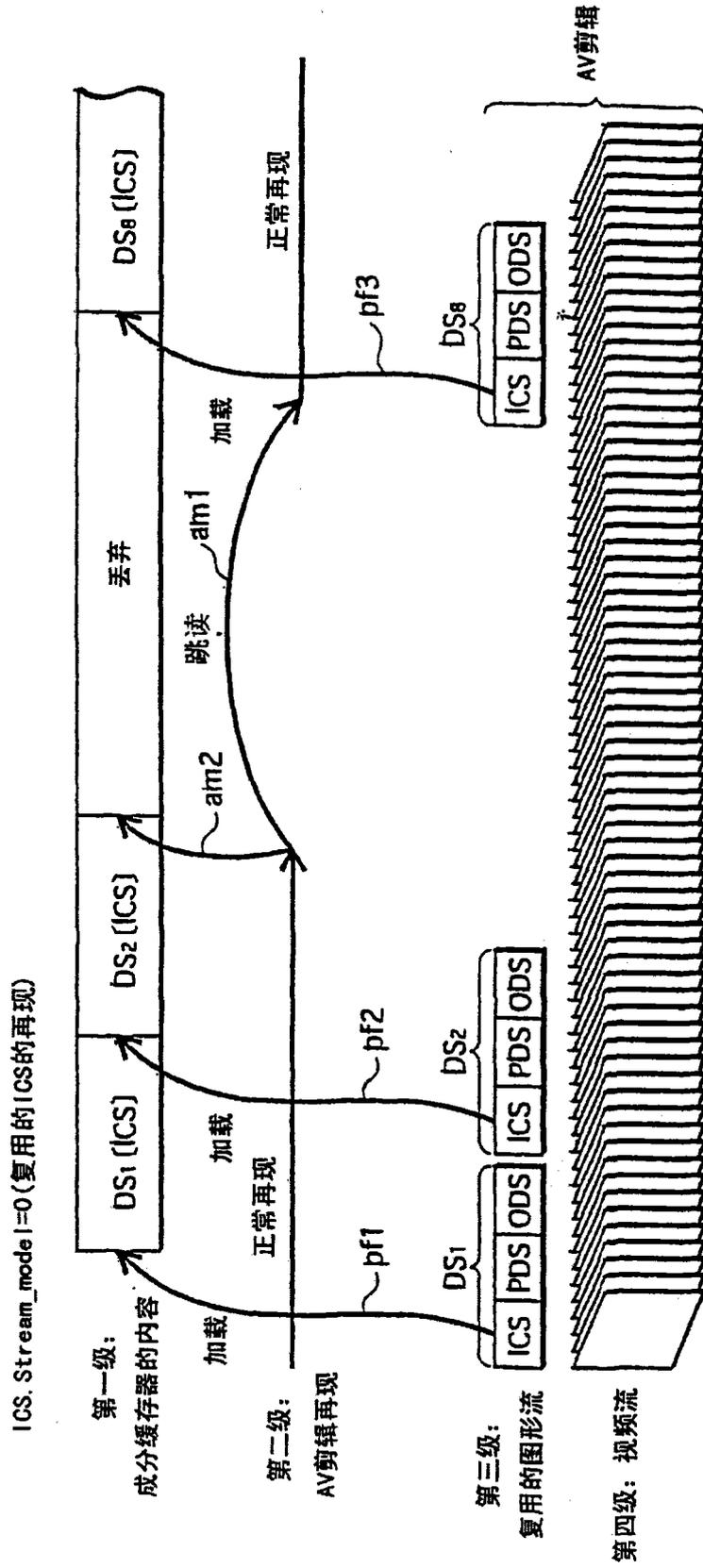


图 9

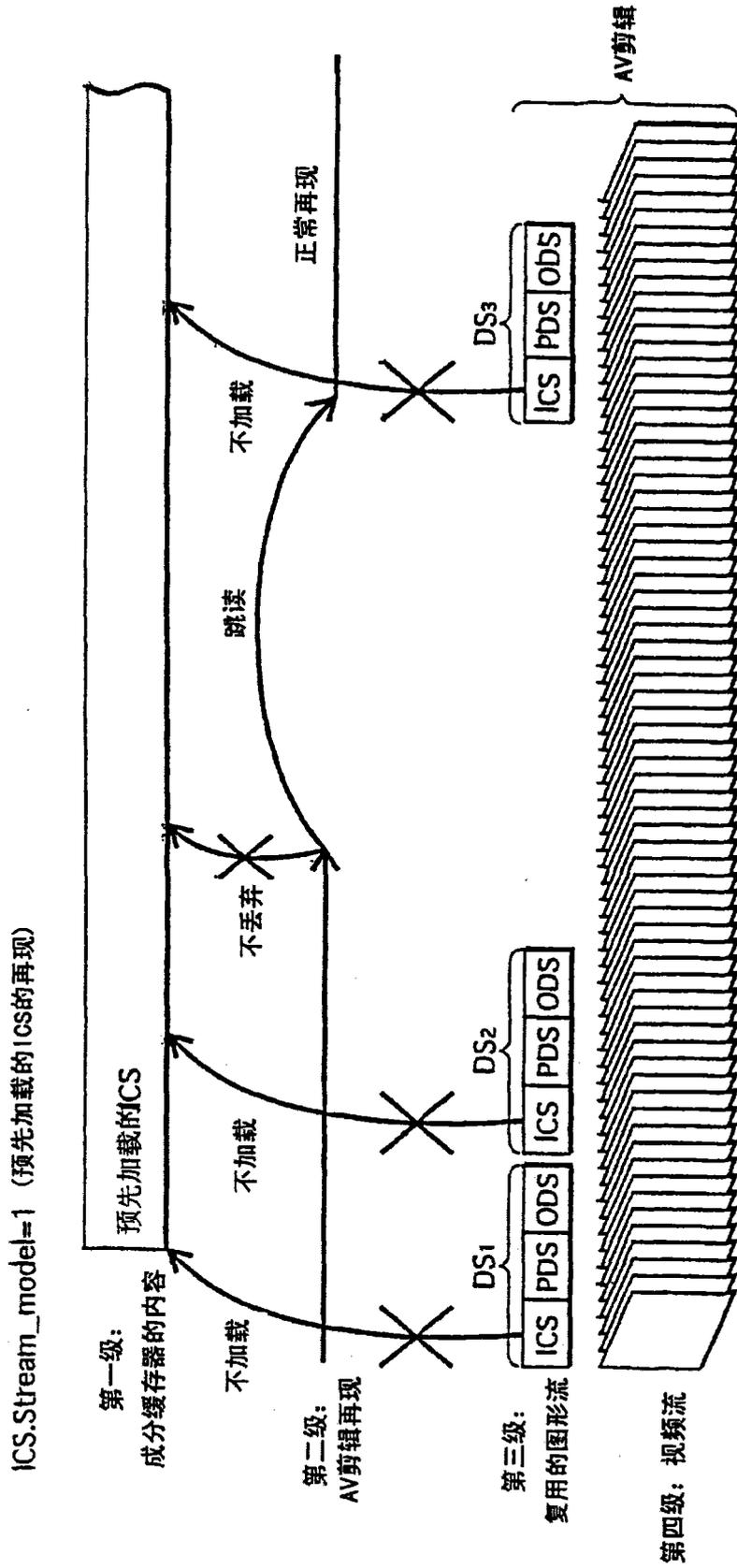


图 10

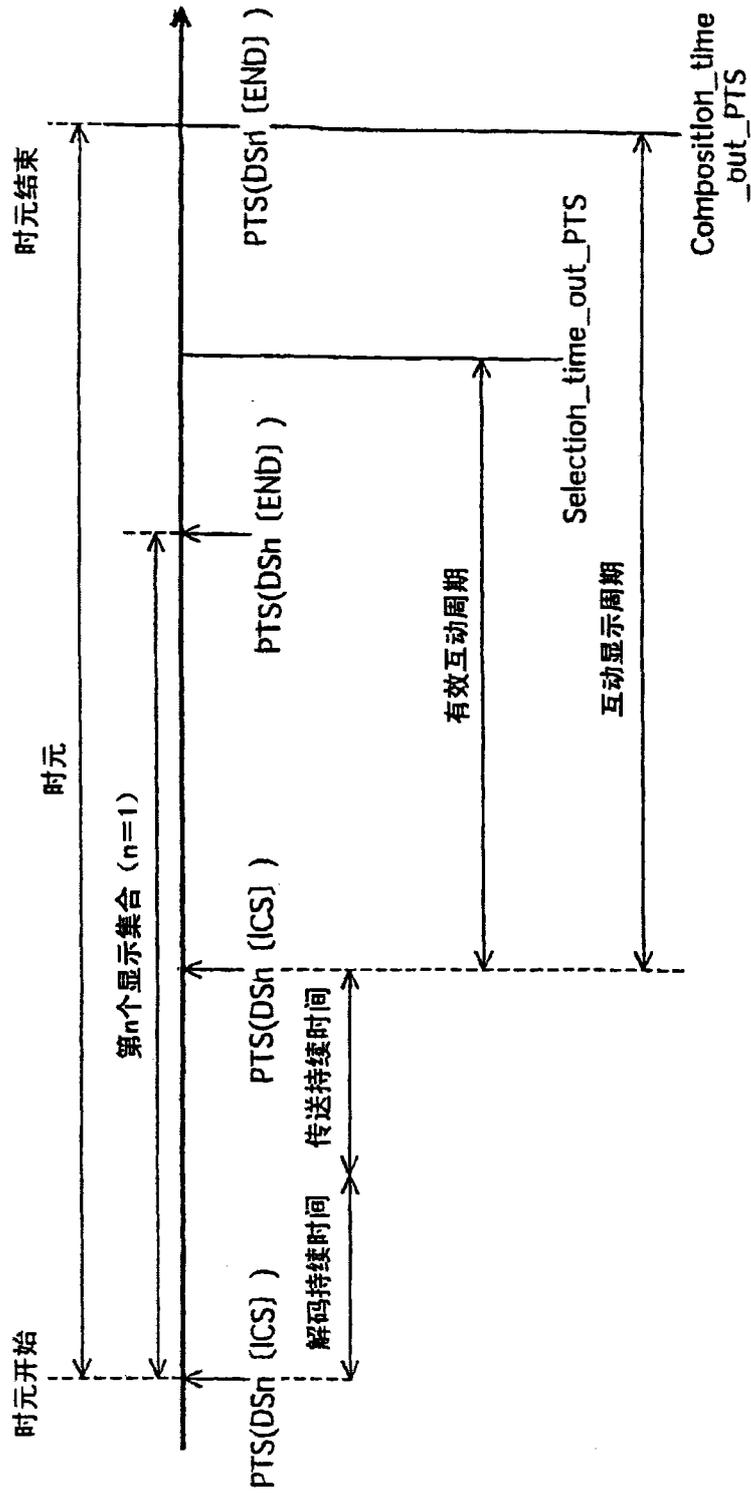


图 11

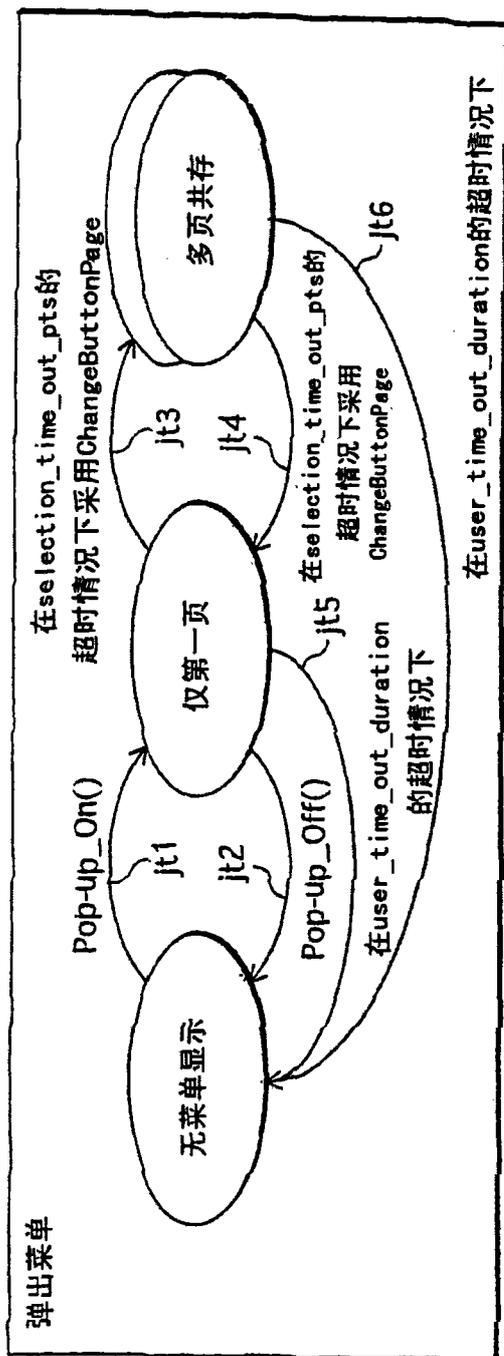


图 12A

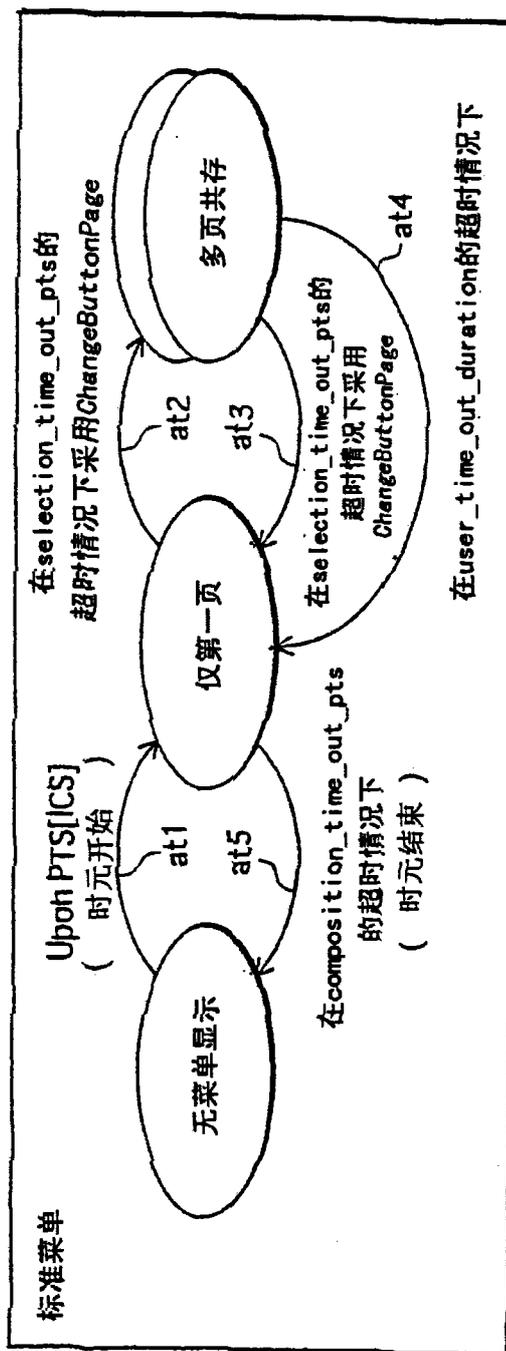


图 12B

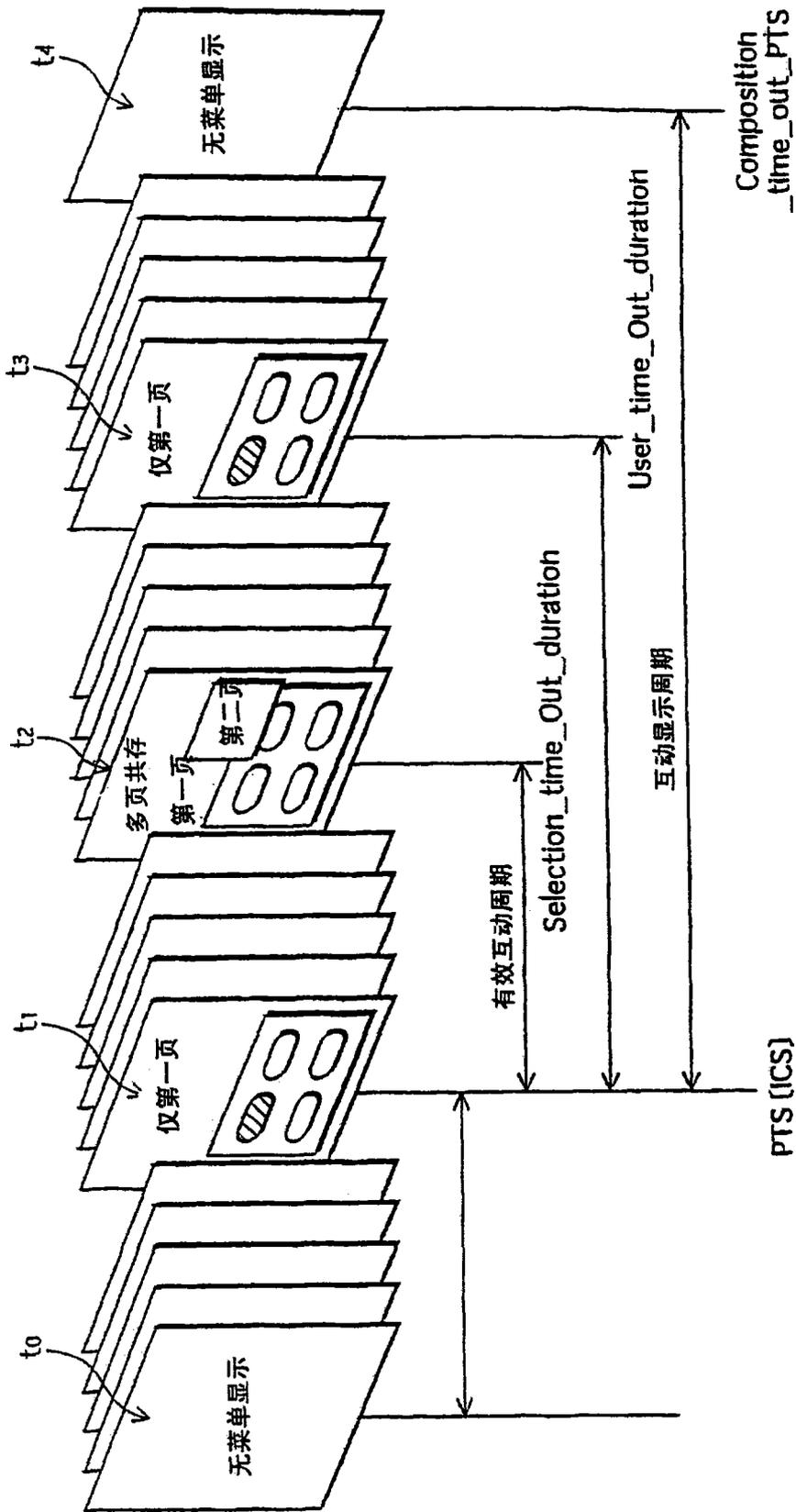


图 13

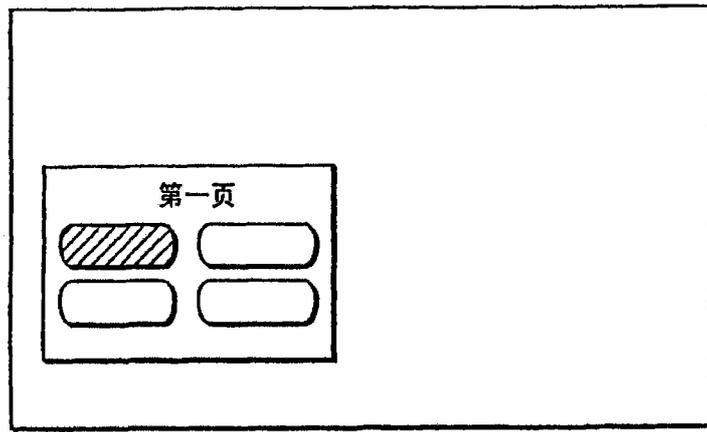


图 14A

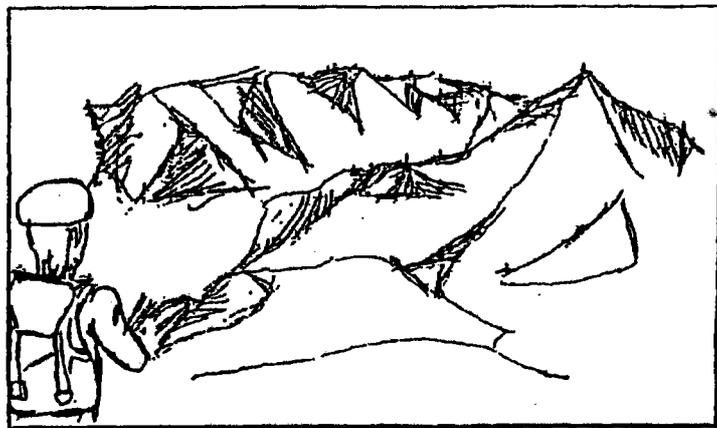


图 14B

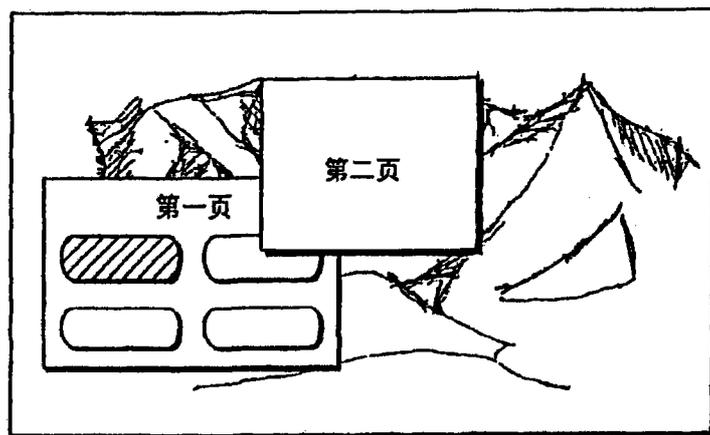
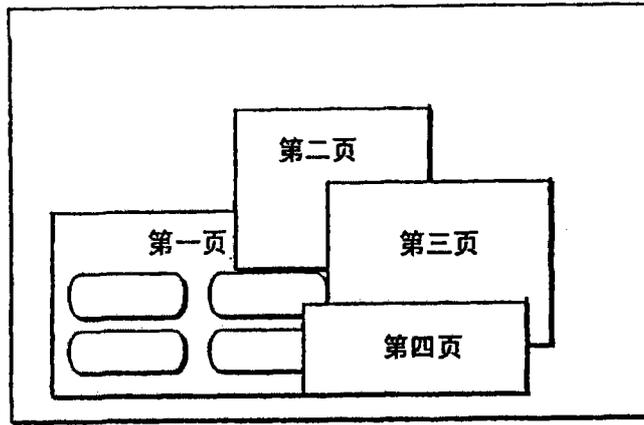


图 14C



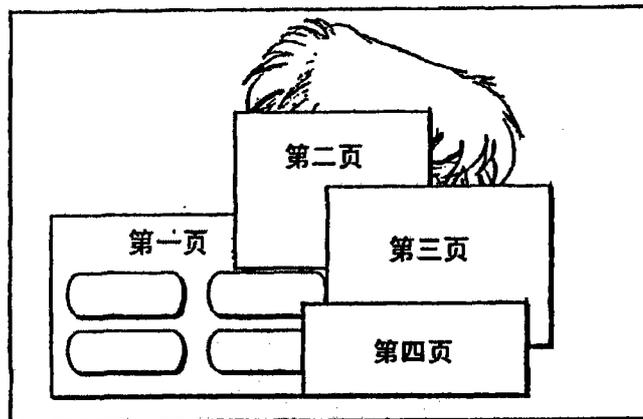
多页共存

图 15A



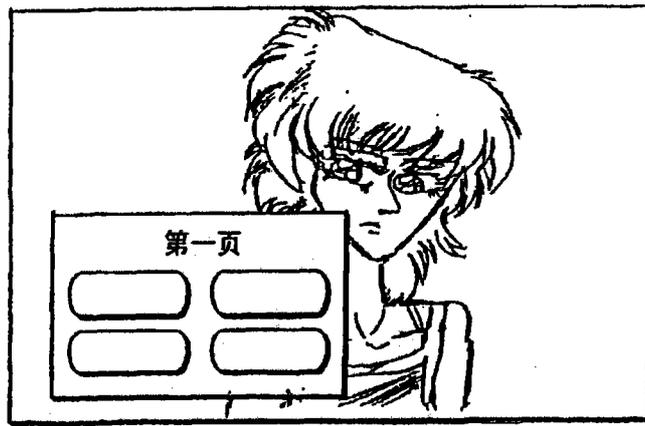
下一画面

图 15B



画面被菜单页覆盖

图 15C



在即将呈现画面之前设置user_time_out_duration

图 15D

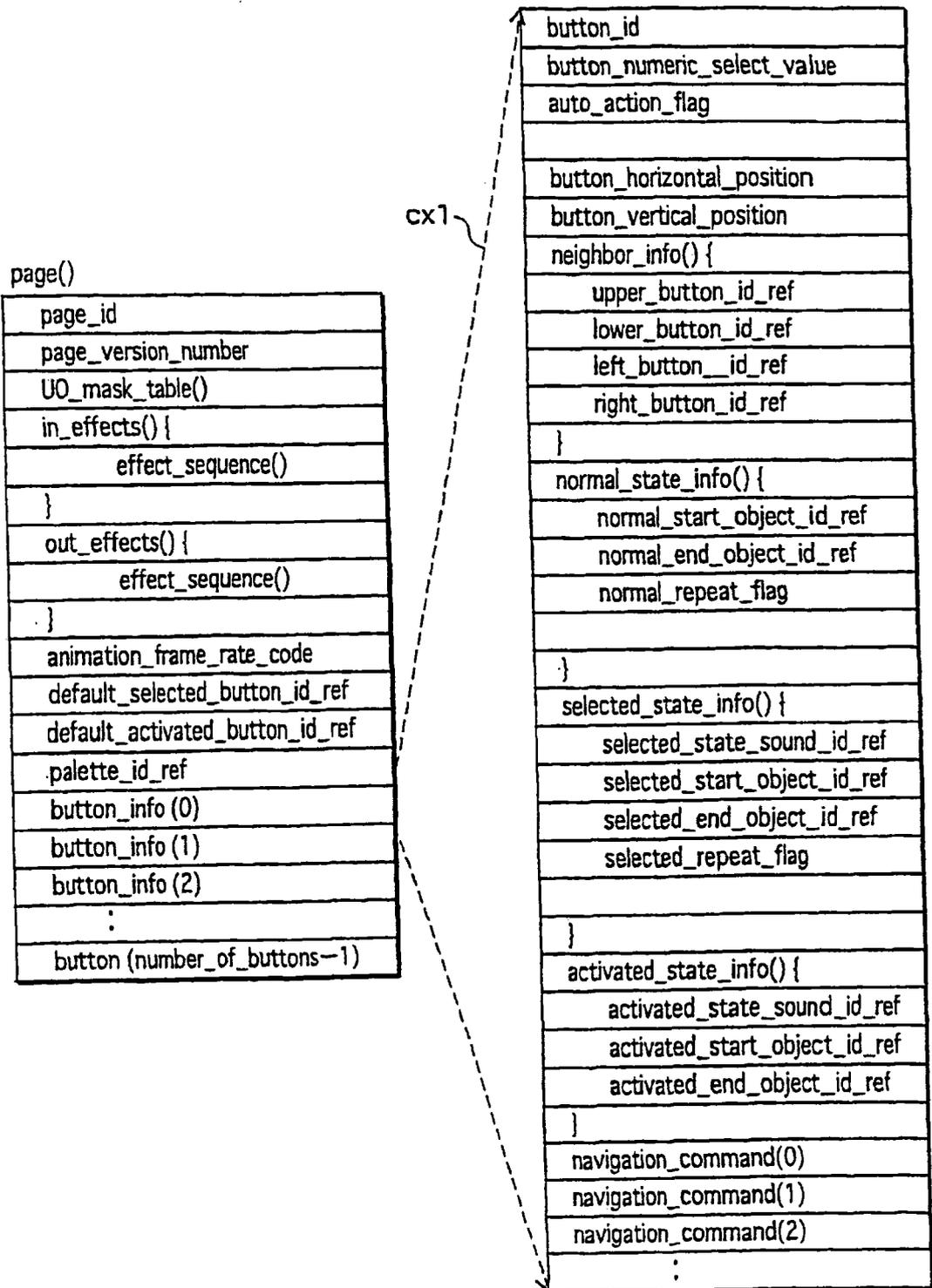


图 16

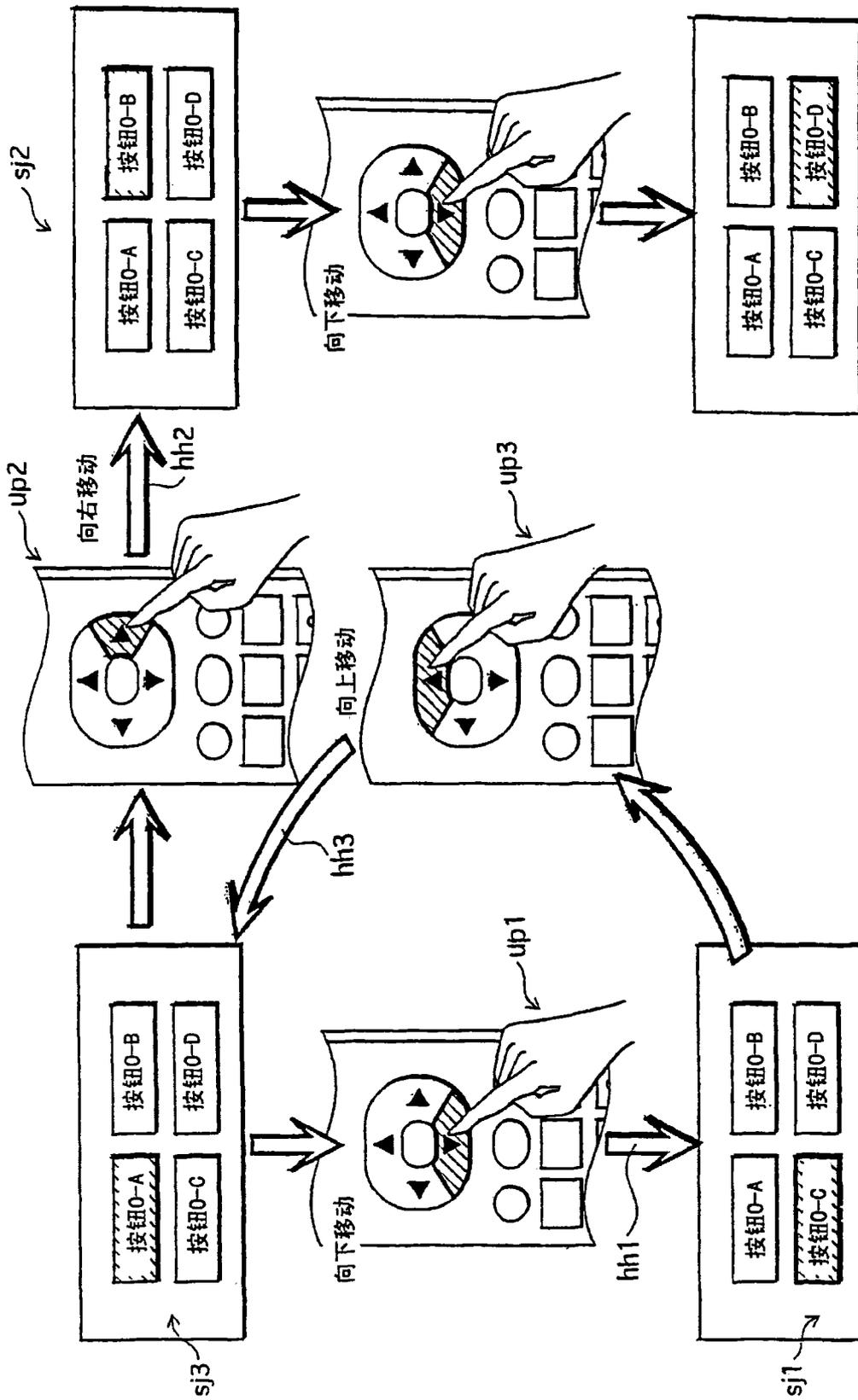


图 17

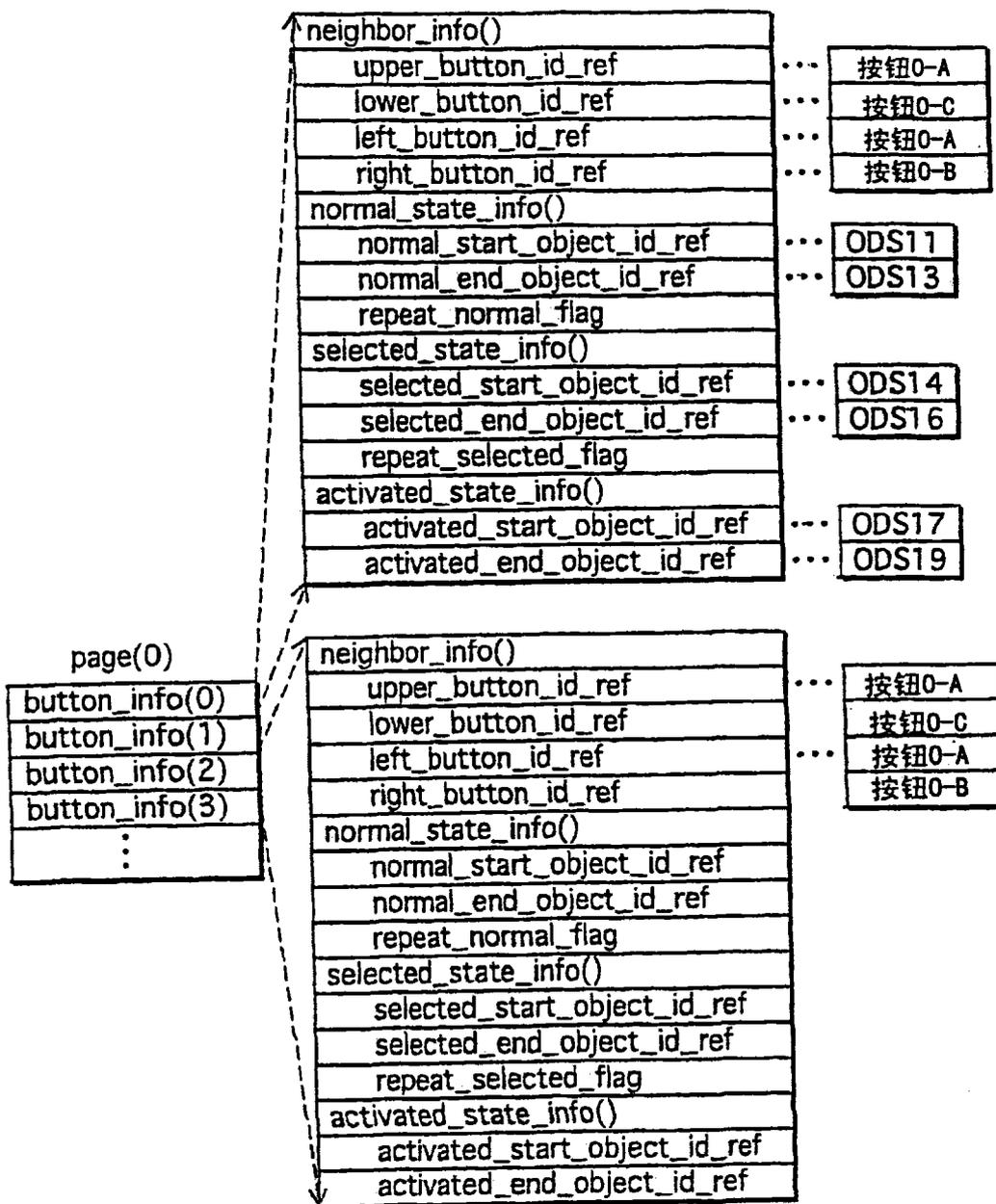


图 18

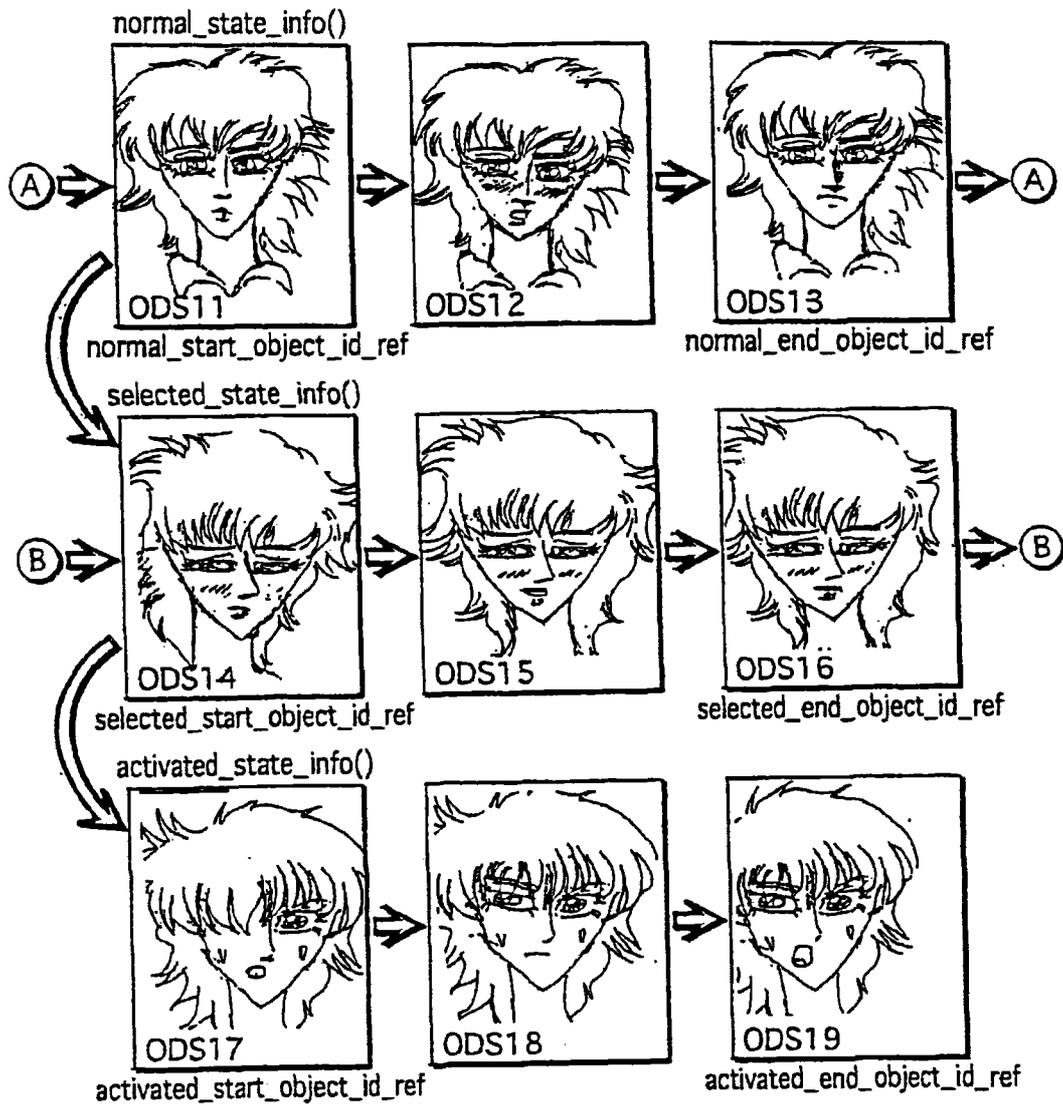


图 19

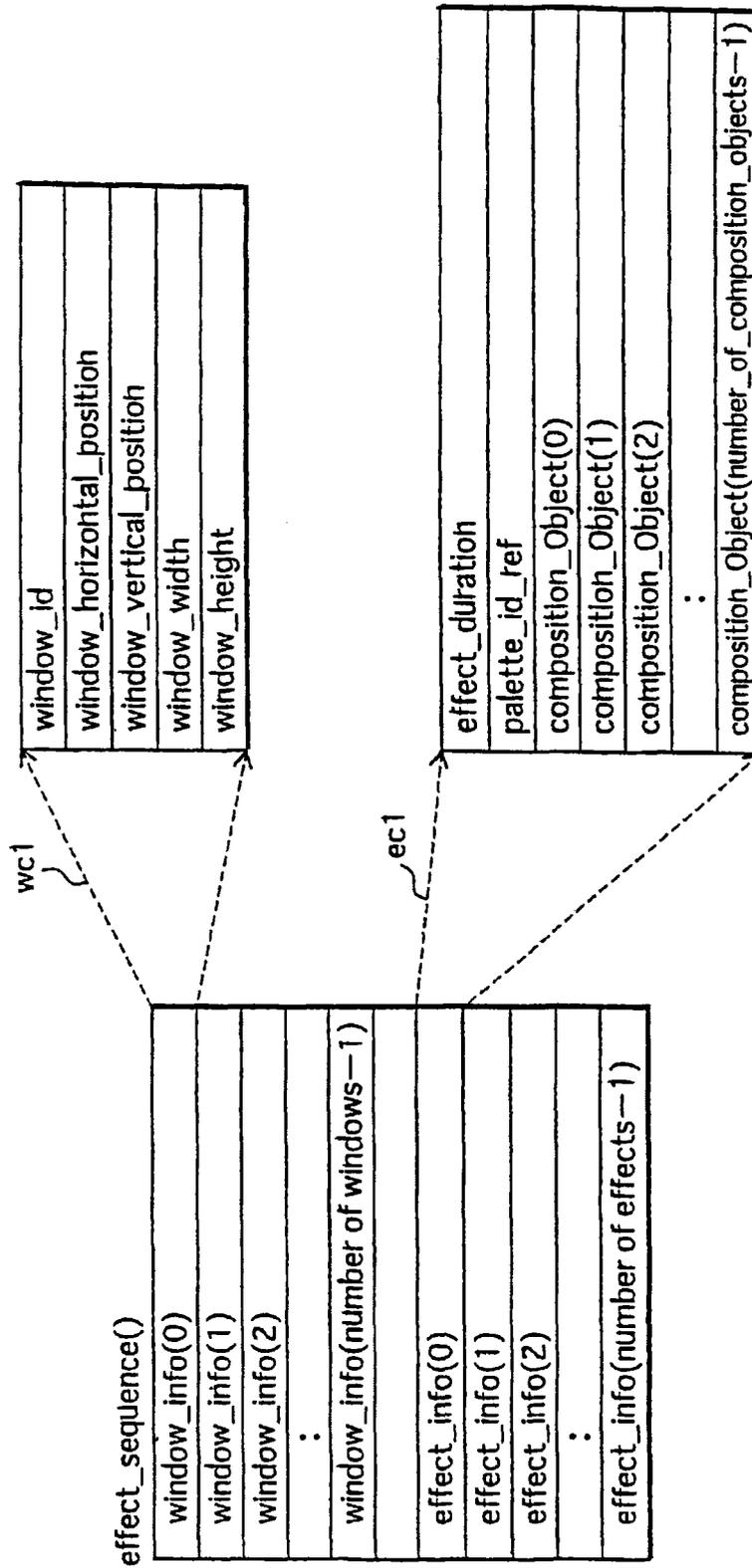


图 20

effect_sequence中的第j个显示成分
(由effect(j)规定的显示成分)

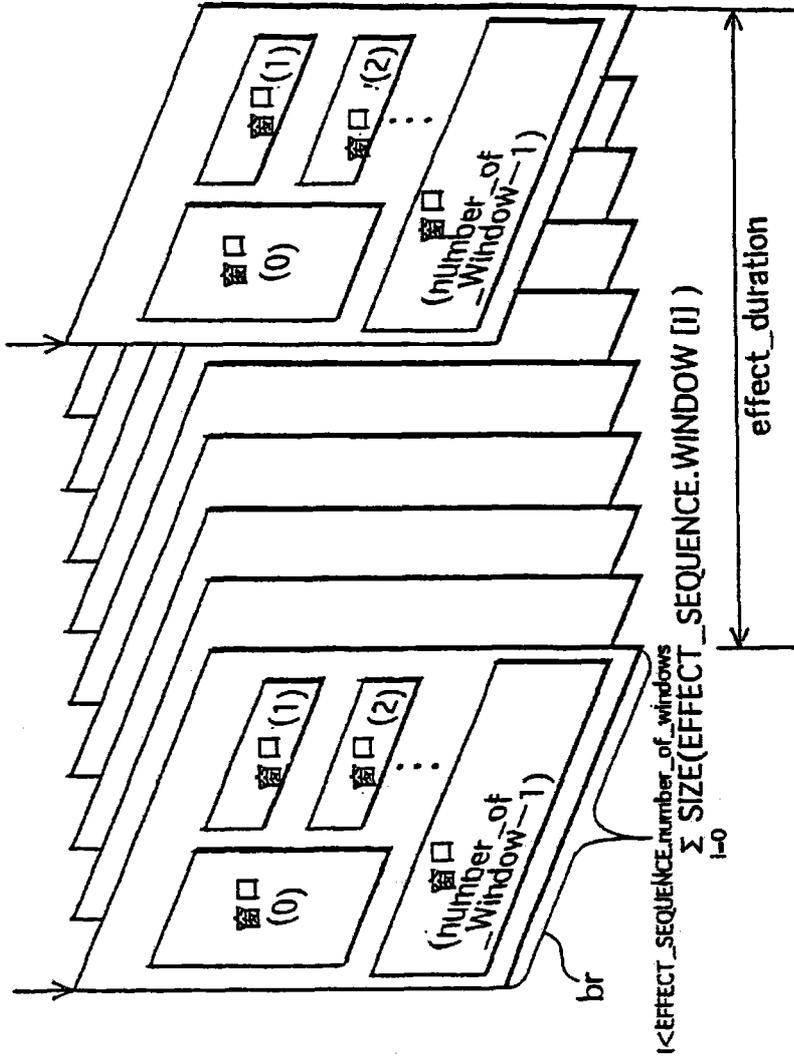


图21A

effect_sequence中的第j+1个显示成分
(由effect(j+1)规定的显示成分)

$$\text{effect_duration} \geq \text{cell} \left(\left(90000 * \left(\sum_{i=0}^{\text{EFFECT_SEQUENCE.number_of_windows}} \text{SIZE}(\text{EFFECT_SEQUENCE.WINDOW [i]}) \right) / (128 * 10^6) \right) \right)$$

图21B

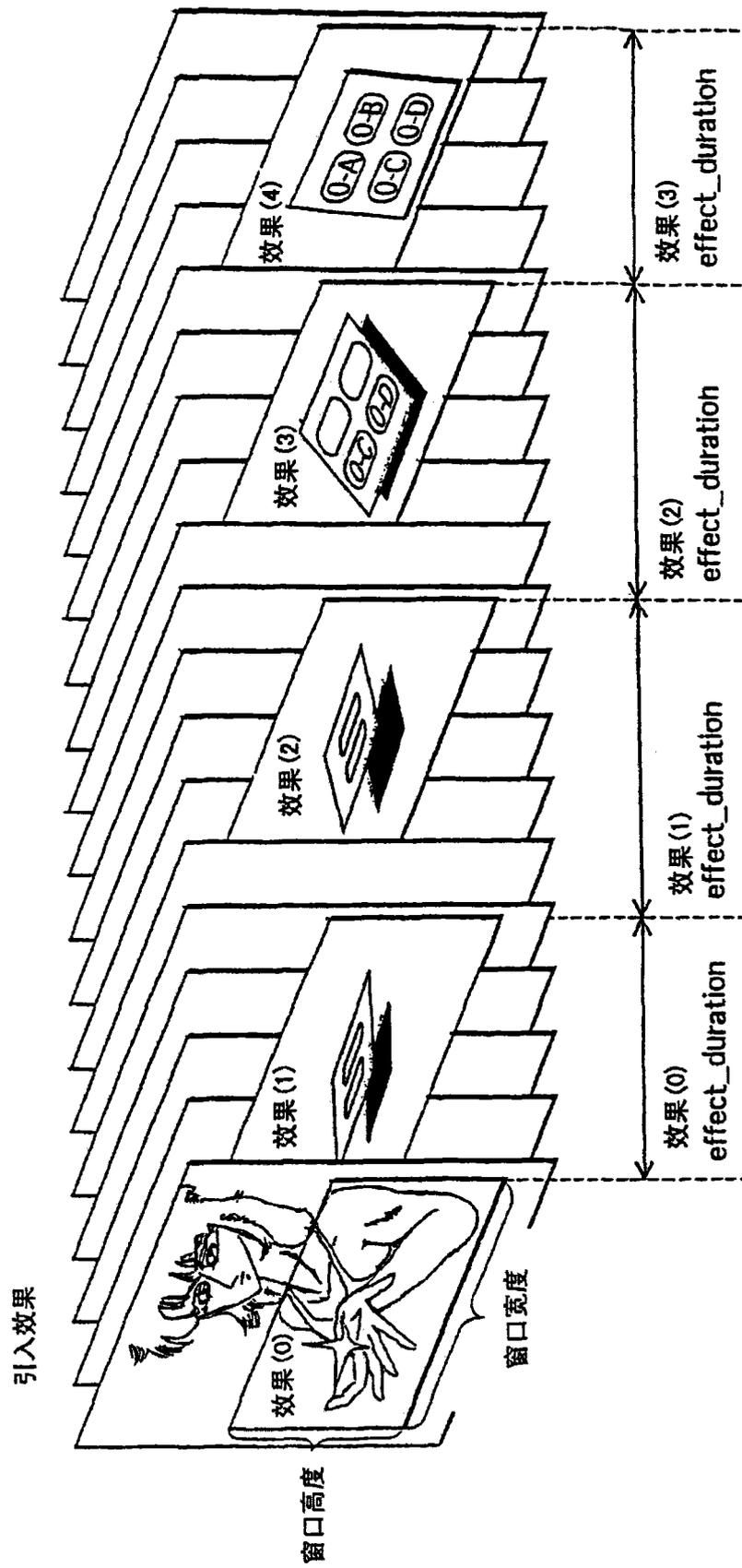


图 22

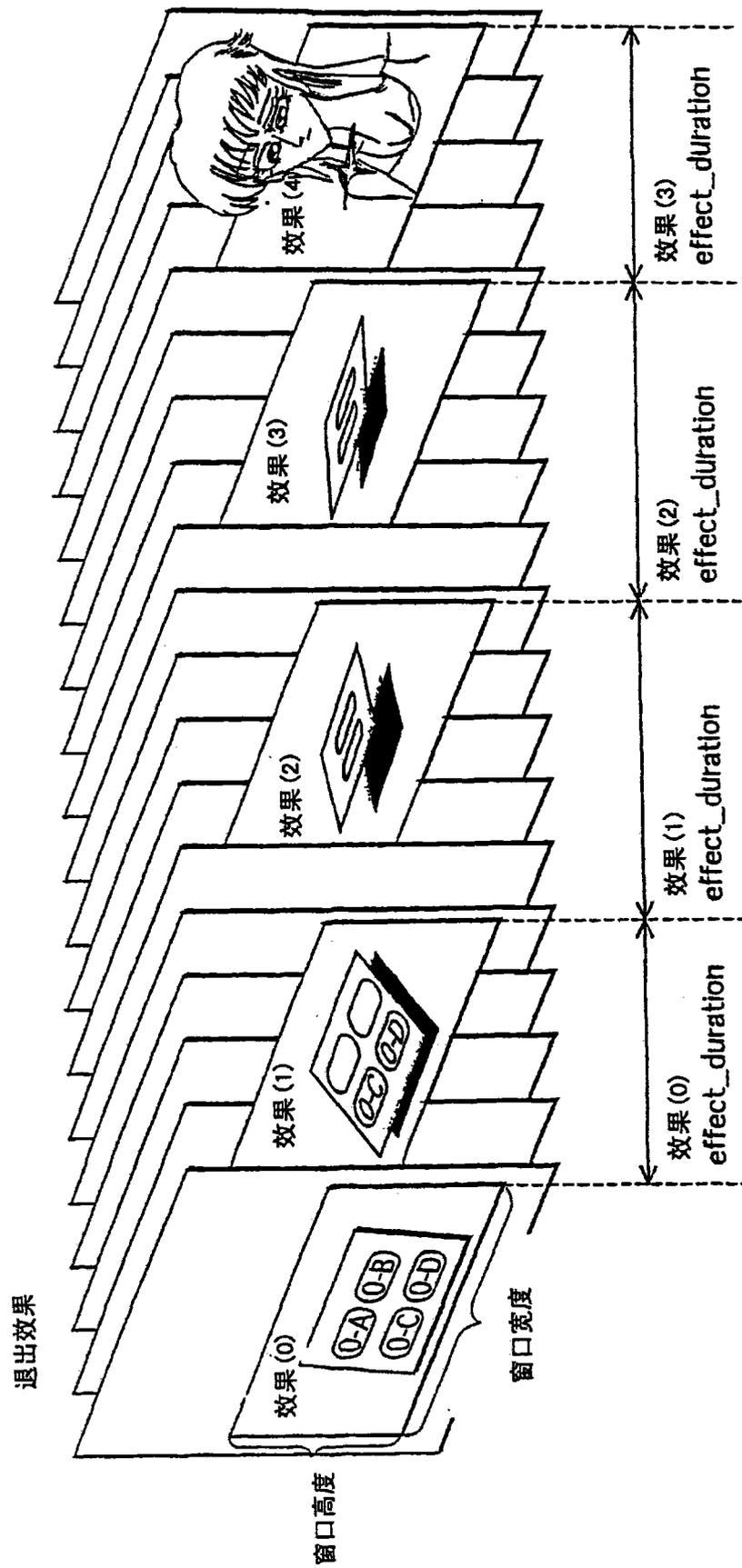


图 23

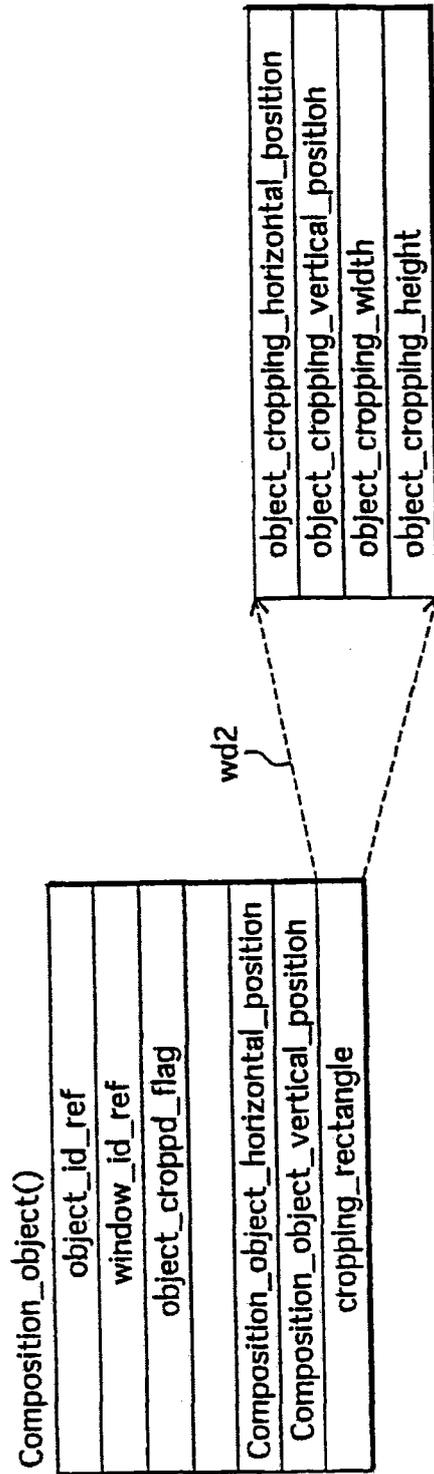


图 24

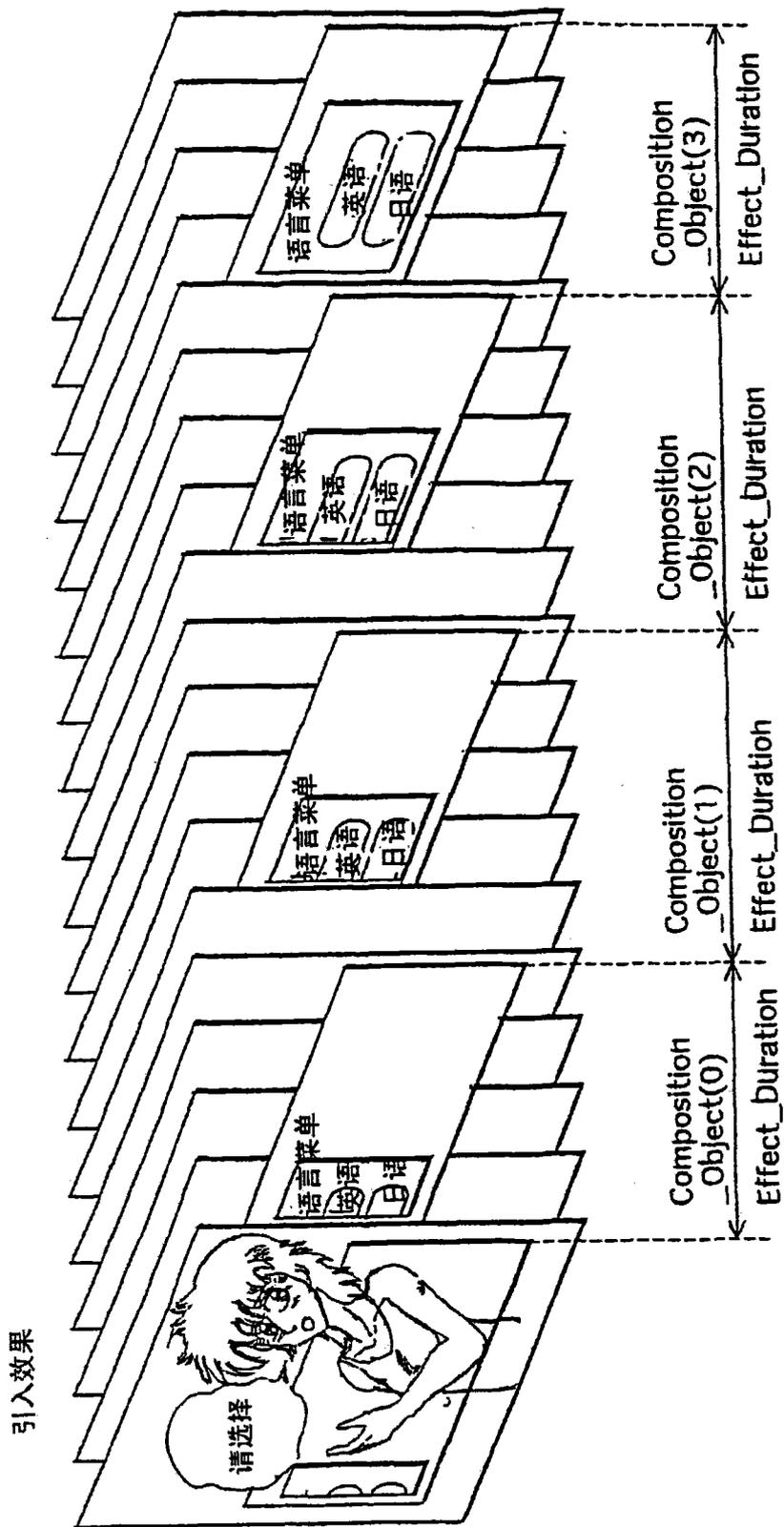


图 25

Composition_Object(0) 设置

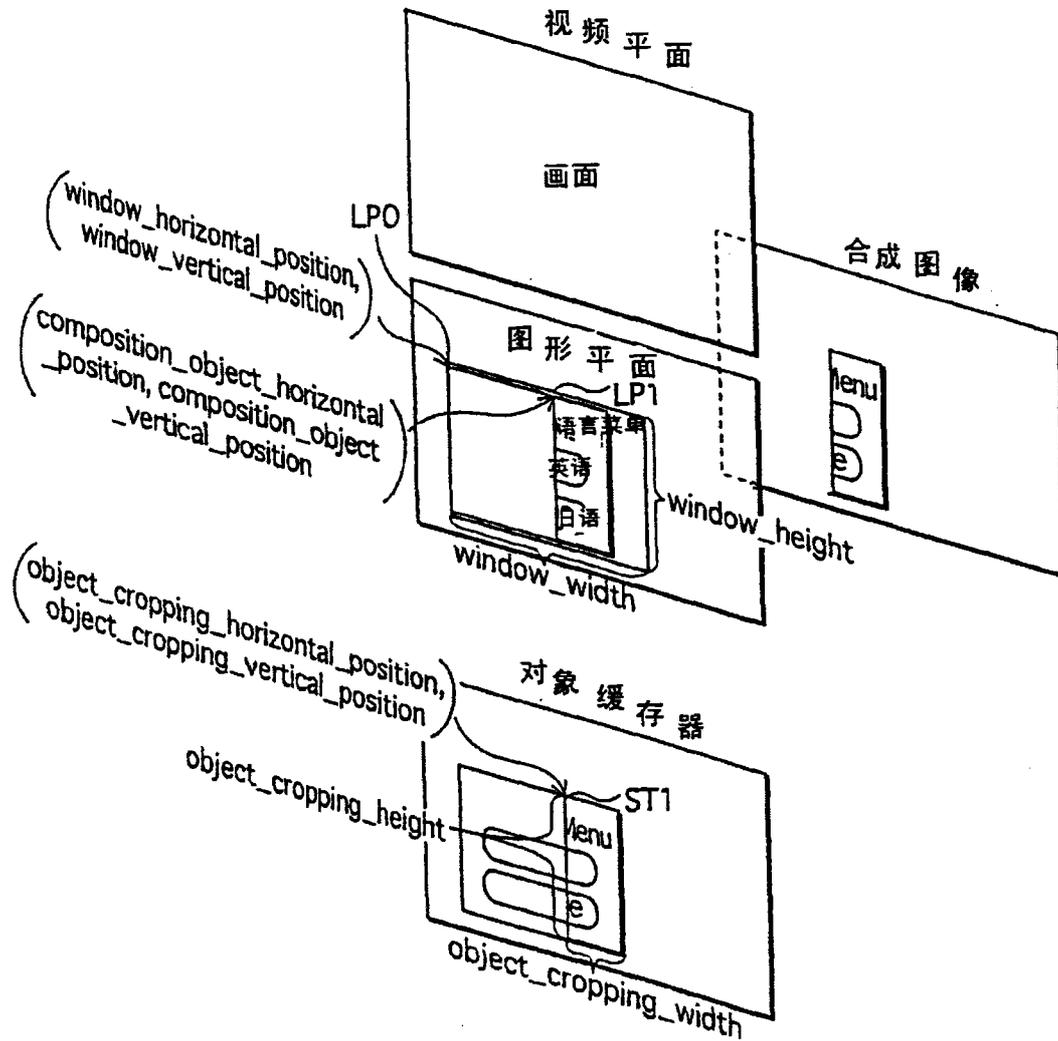


图 26

Composition_Object(1) 设置

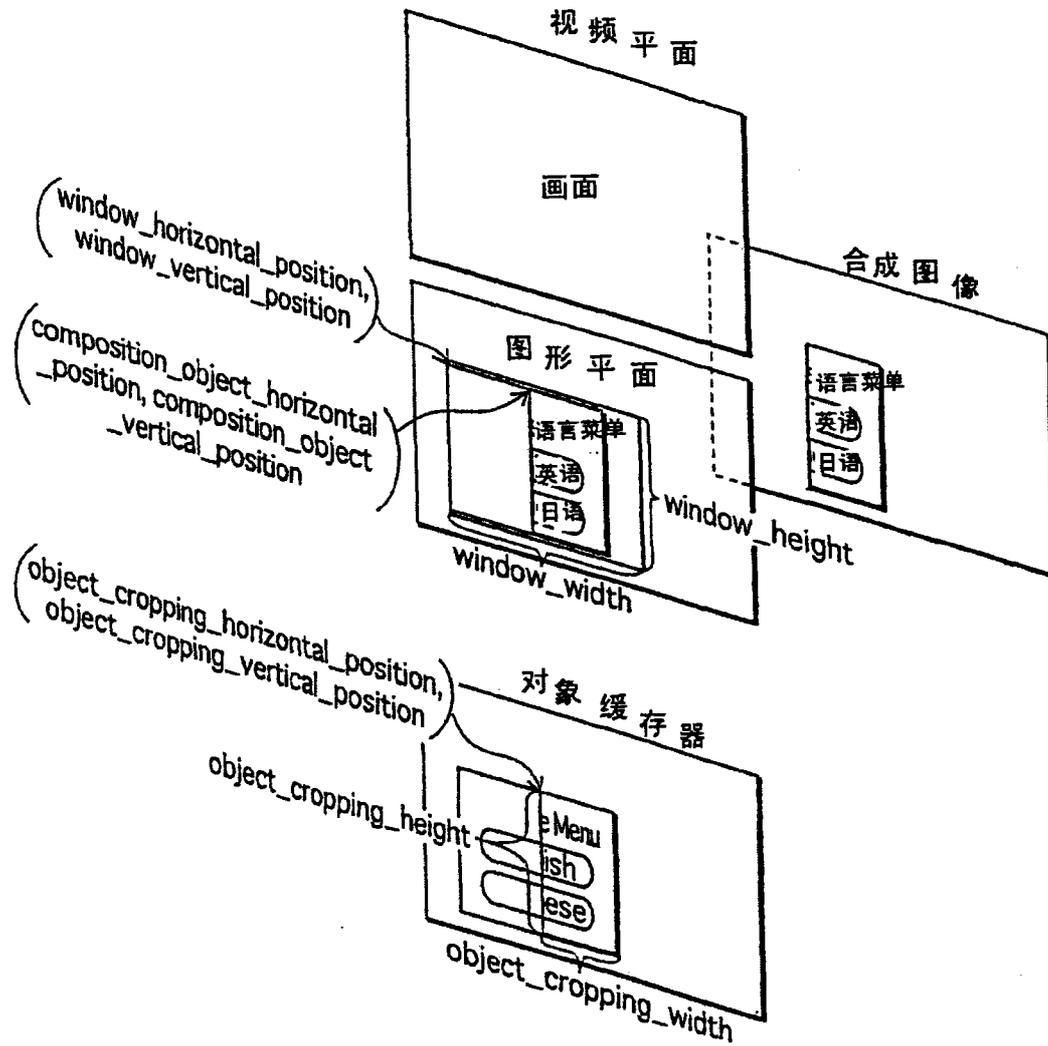


图 27

Composition_Object(2) 设置

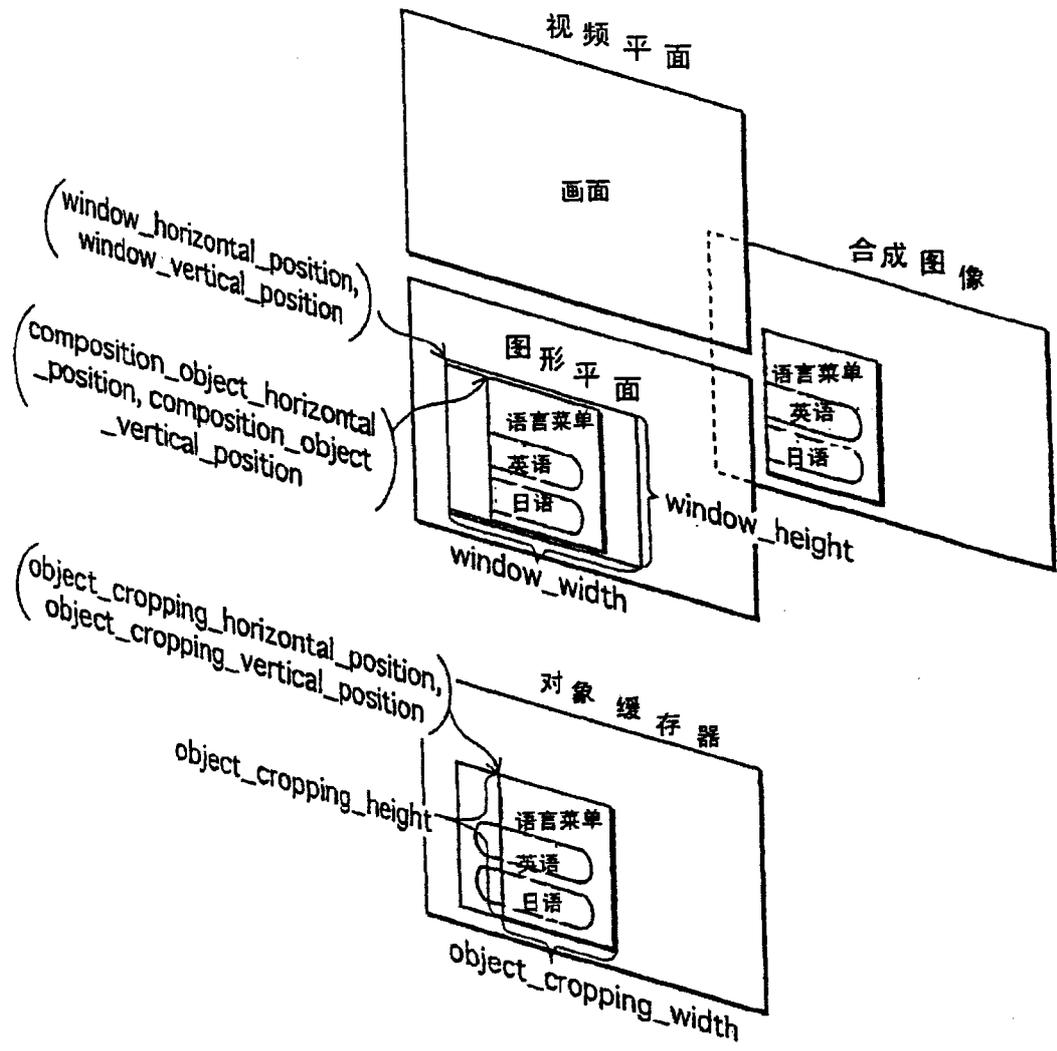


图 28

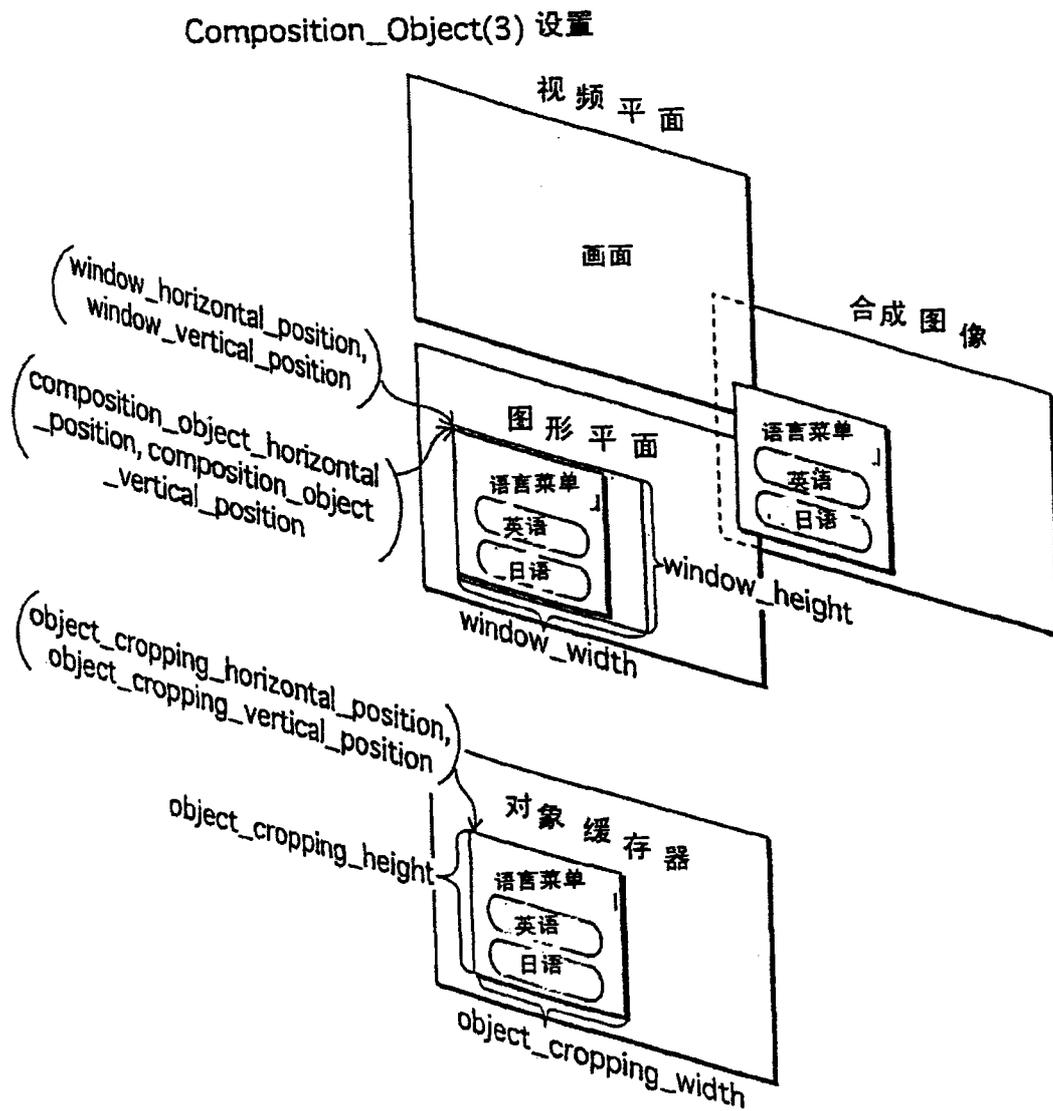


图 29

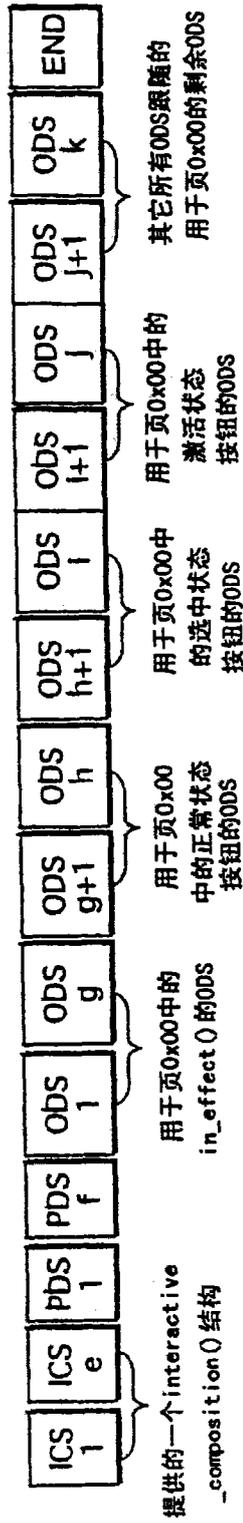


图 30

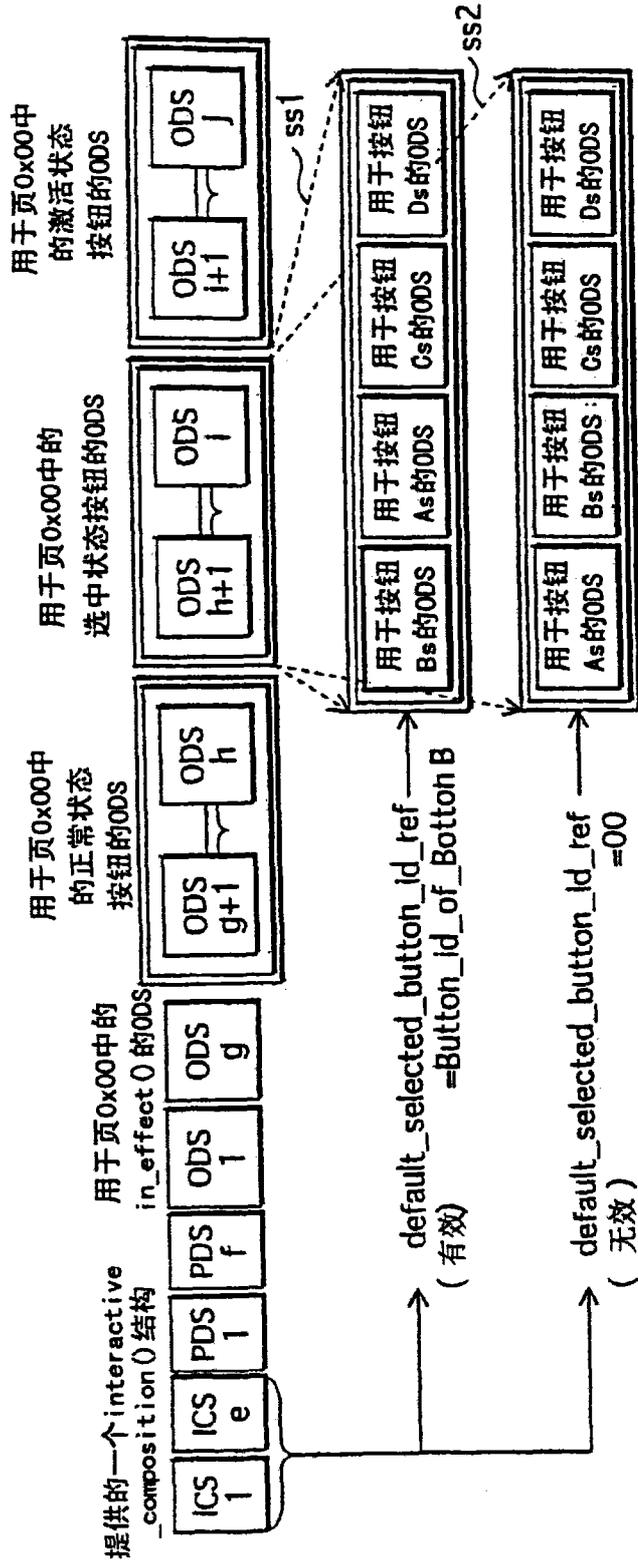


图 31

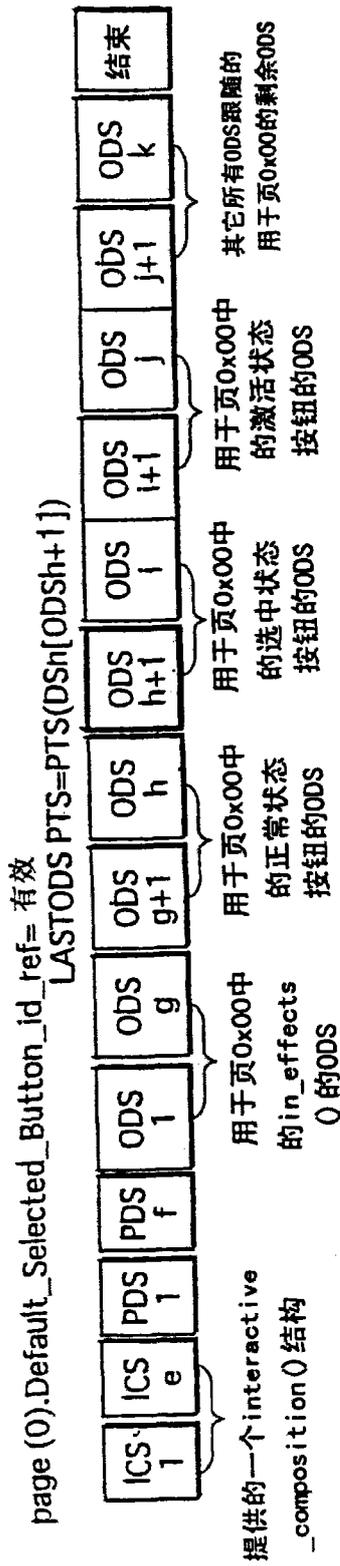


图 32A

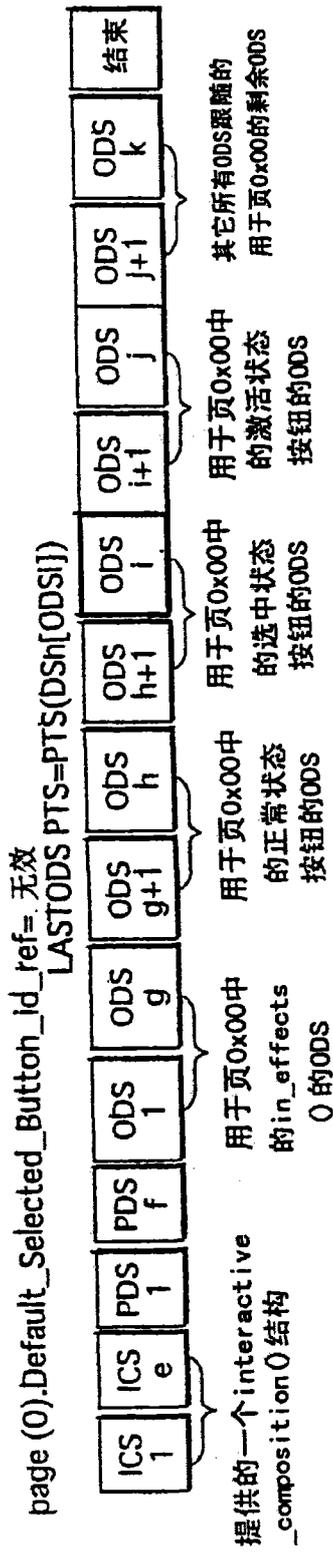


图 32B

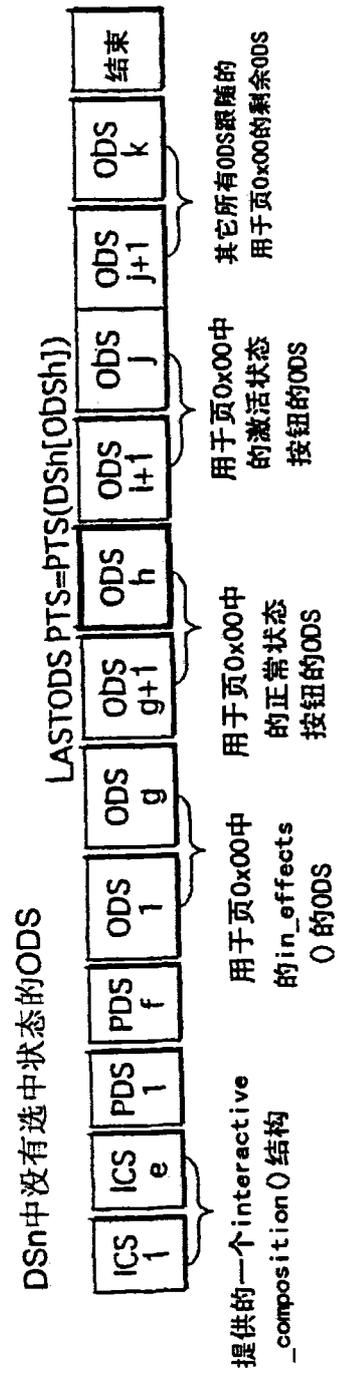


图 33A

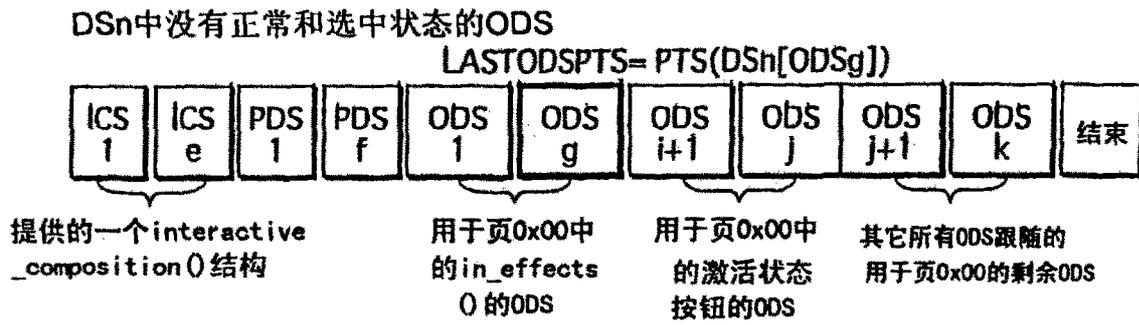


图 33B

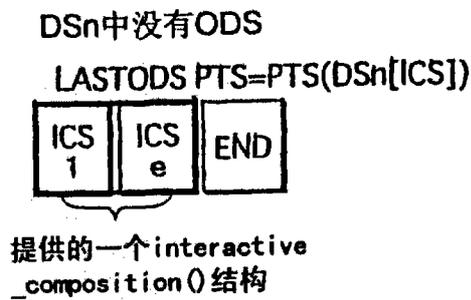


图 33C

$$PTS(DSn[ICS]) \geq DTS(DSn[ICS]) + DECODEDURATION(DSn) + TRANSFERDURATION(DSn)$$

其中：

按照下述方法计算 `DECODEDURATION(DSn)`

```

if( DSn[ICS]. composition_state == EPOCH_START )
    return(max( LASTODSPTS(DSn) - DTS(DSn[ICS]), PLANECLARTIME(DSn) )
else
    return(LASTODSPTS(DSn) - DTS(DSn[ICS])
    
```

图 34A

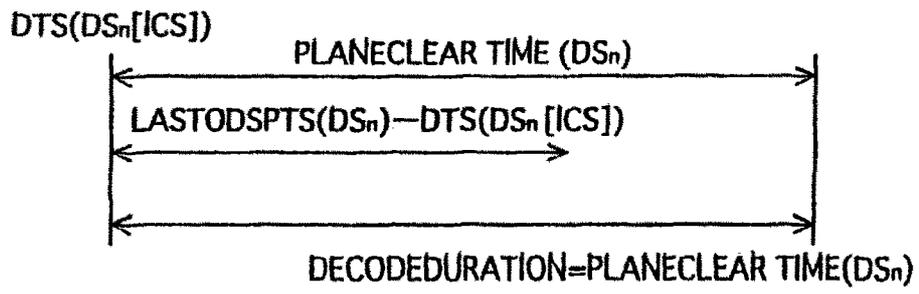


图 34B

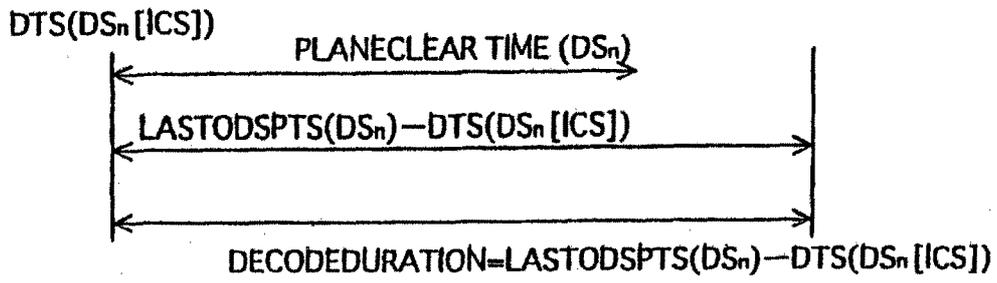


图 34C

其中：

$$\text{EFFECTTD}(DS_n) = \text{ceil} \left(\left(\sum_{i=0}^{i < \text{ICS.PAGE}[0].\text{IN_EFFECTS.number_of_windows}} \text{SIZE}(DS_n[\text{ICS}].\text{PAGE}[0].\text{IN_EFFECTS.WINDOW}[i]) \right) / (128 * 10^6) \right)$$

图 35A

$$\text{PAGEDEFAULTTD}(DS_n) = \text{ceil} \left(\left(\sum_{i=0}^{i < \text{ICS.PAGE}[0].\text{number_of_button}} \text{NBSIZE}(DS_n, DS_n[\text{ICS}].\text{PAGE}[0].\text{button}) - \text{NBSIZE}(DS_n, DS_n[\text{ICS}].\text{PAGE}[0].\text{default_selected_button_id_ref}) + \text{SBSIZE}(DS_n, DS_n[\text{ICS}].\text{PAGE}[0].\text{default_selected_button_id_ref}) \right) / (128 * 10^6) \right)$$

图 35B

$$\text{PAGENODEFAULTTD}(DS_n) = \text{ceil} \left(\left(\sum_{i=0}^{i < \text{ICS.PAGE}[0].\text{number_of_button}} \text{NBSIZE}(DS_n, DS_n[\text{ICS}].\text{PAGE}[0].\text{button}) + \text{BSIZE}(DS_n, \text{LRG}\{\text{button} : \text{button} \in DS_n[\text{ICS}].\text{PAGE}[0].\text{button}\}) - \text{NBSIZE}(DS_n, \text{LRG}\{\text{button} : \text{button} \in DS_n[\text{ICS}].\text{PAGE}[0].\text{button}\}) \right) / (128 * 10^6) \right)$$

图 35C

$PTS(DS_n [ICS]) \geq DTS(DS_n [ICS]) + DECODEDURATION(DS_n) + TRANSFERDURATION(DS_n)$

其中：

- 按照下述方法计算TRANSFERDURATION (DS_n)

```
if ( DSn[ICS]. PAGE[0]. IN_EFFECTS.number_of_effects != 0 )
    return EFFECTTD(DSn)
else if( DSn[ICS]. PAGE[0]. default_selected_button_id_ref == 0xFFFF )
    return PAGENODEFAULTTD(DSn)
else
    return PAGEDEFAULTTD(DSn)
```

图 36

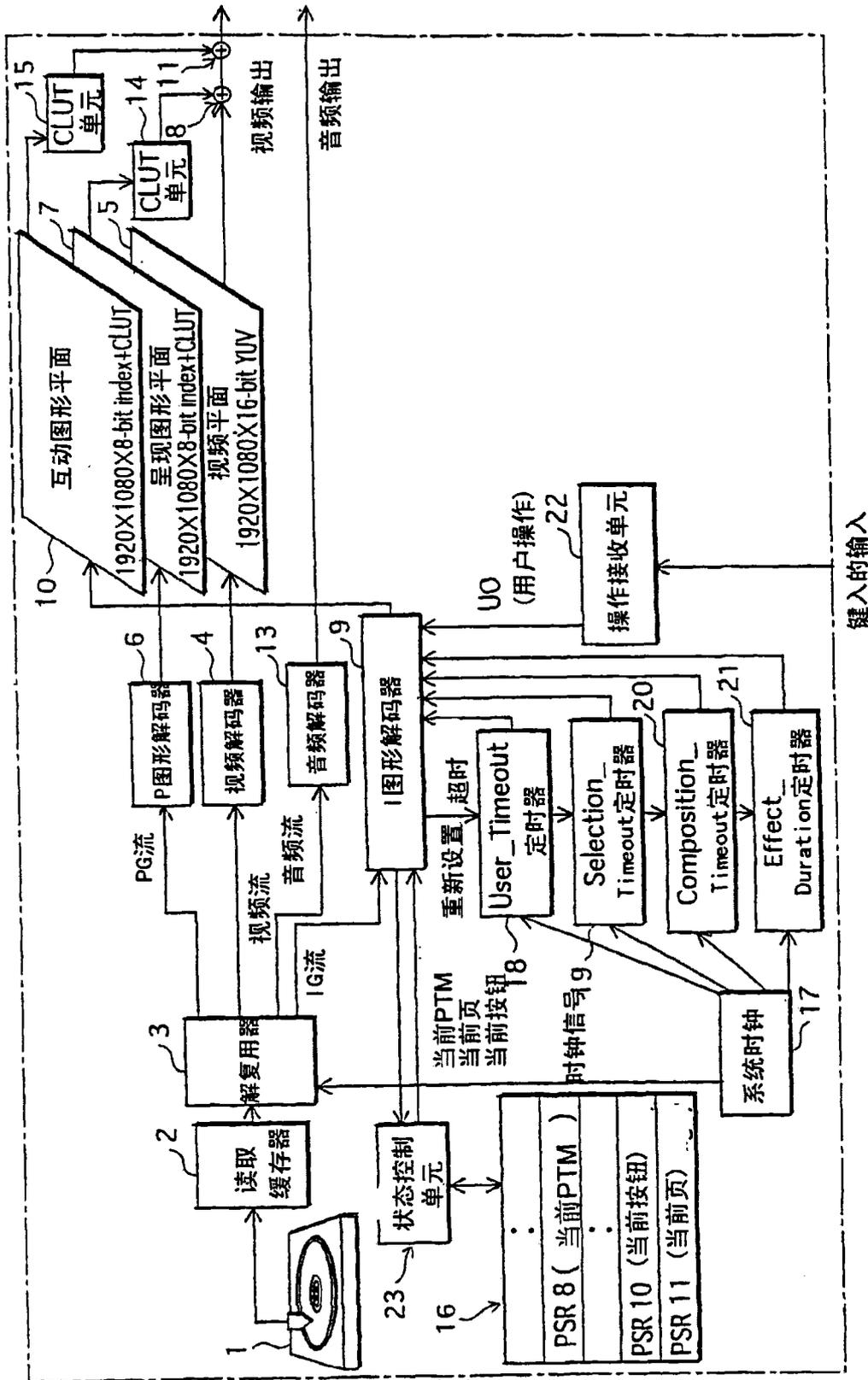


图 37

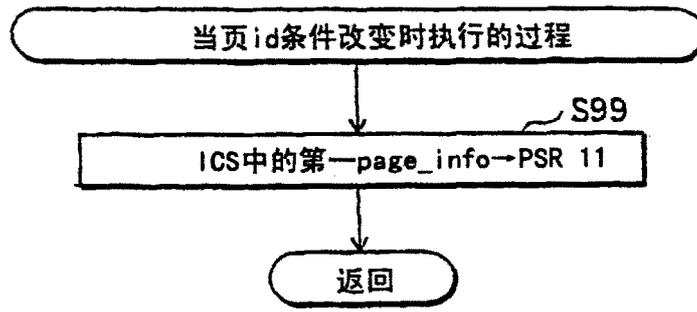


图 38A

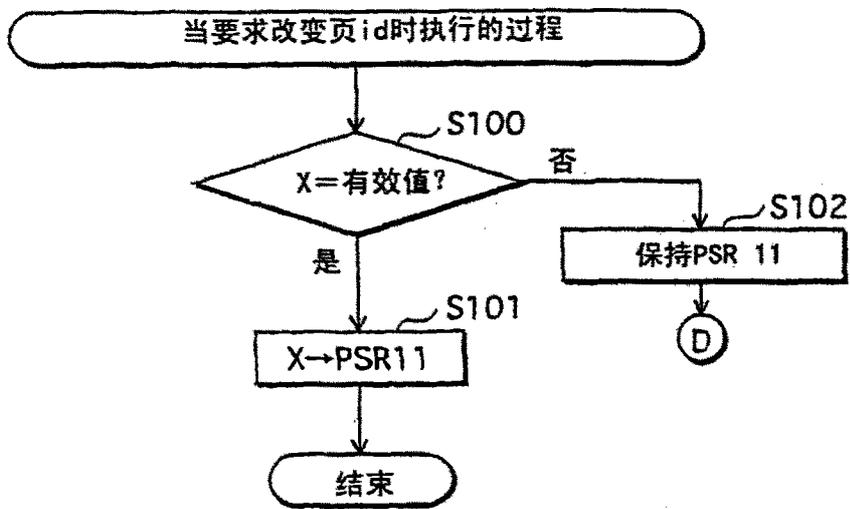


图 38B

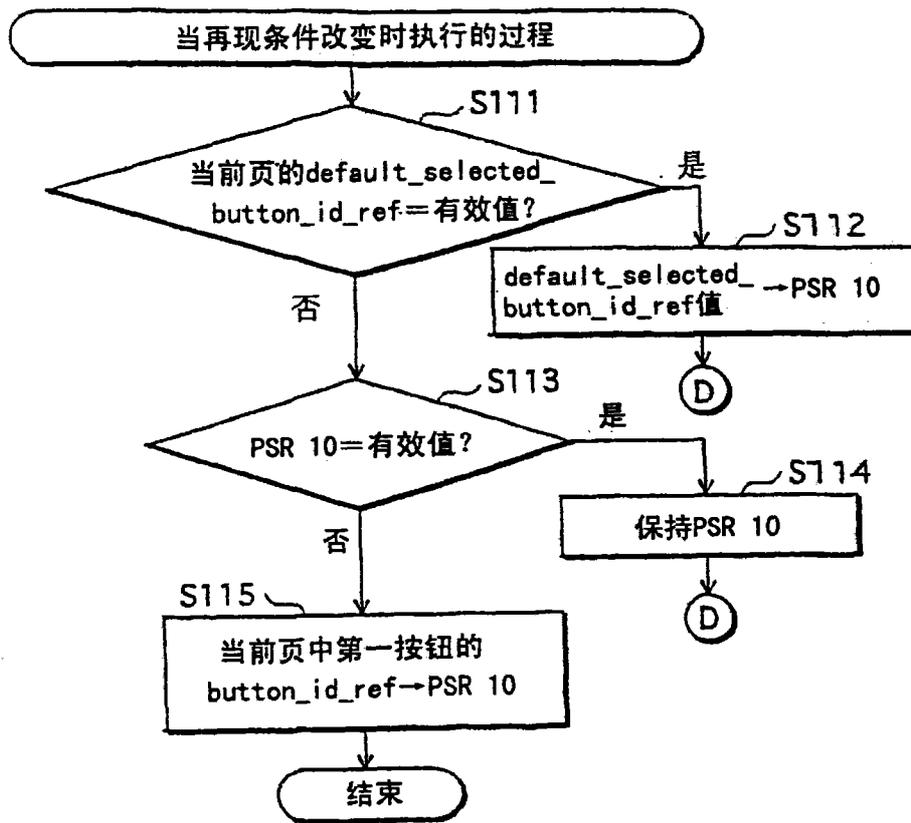


图 39A

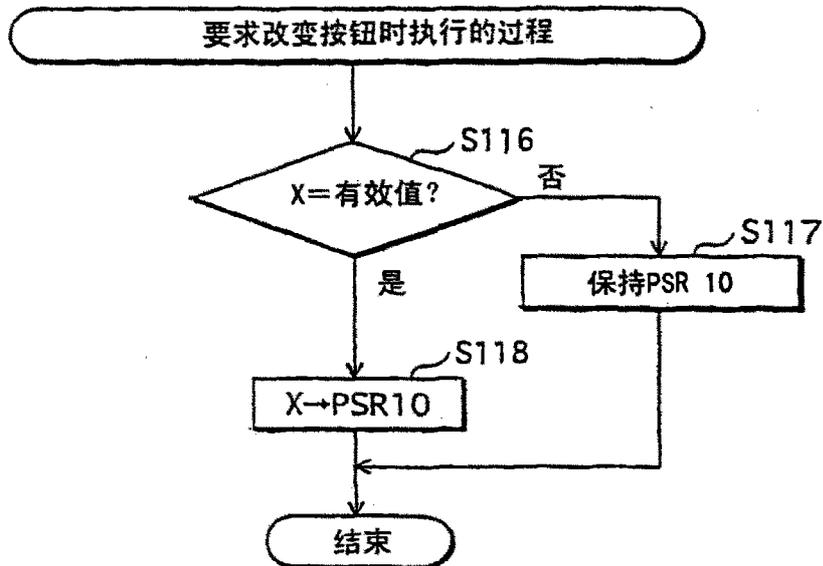


图 39B

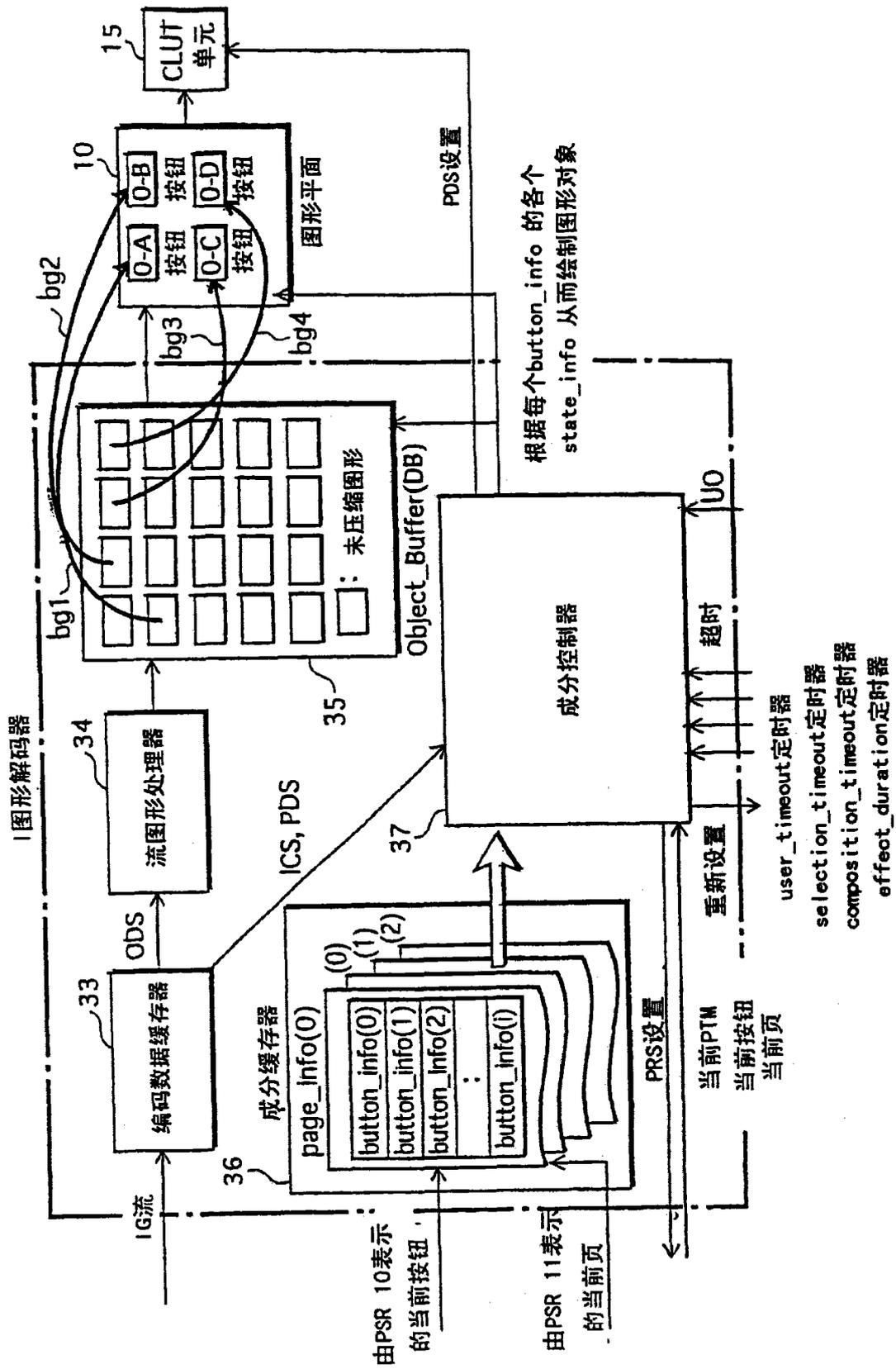


图 40

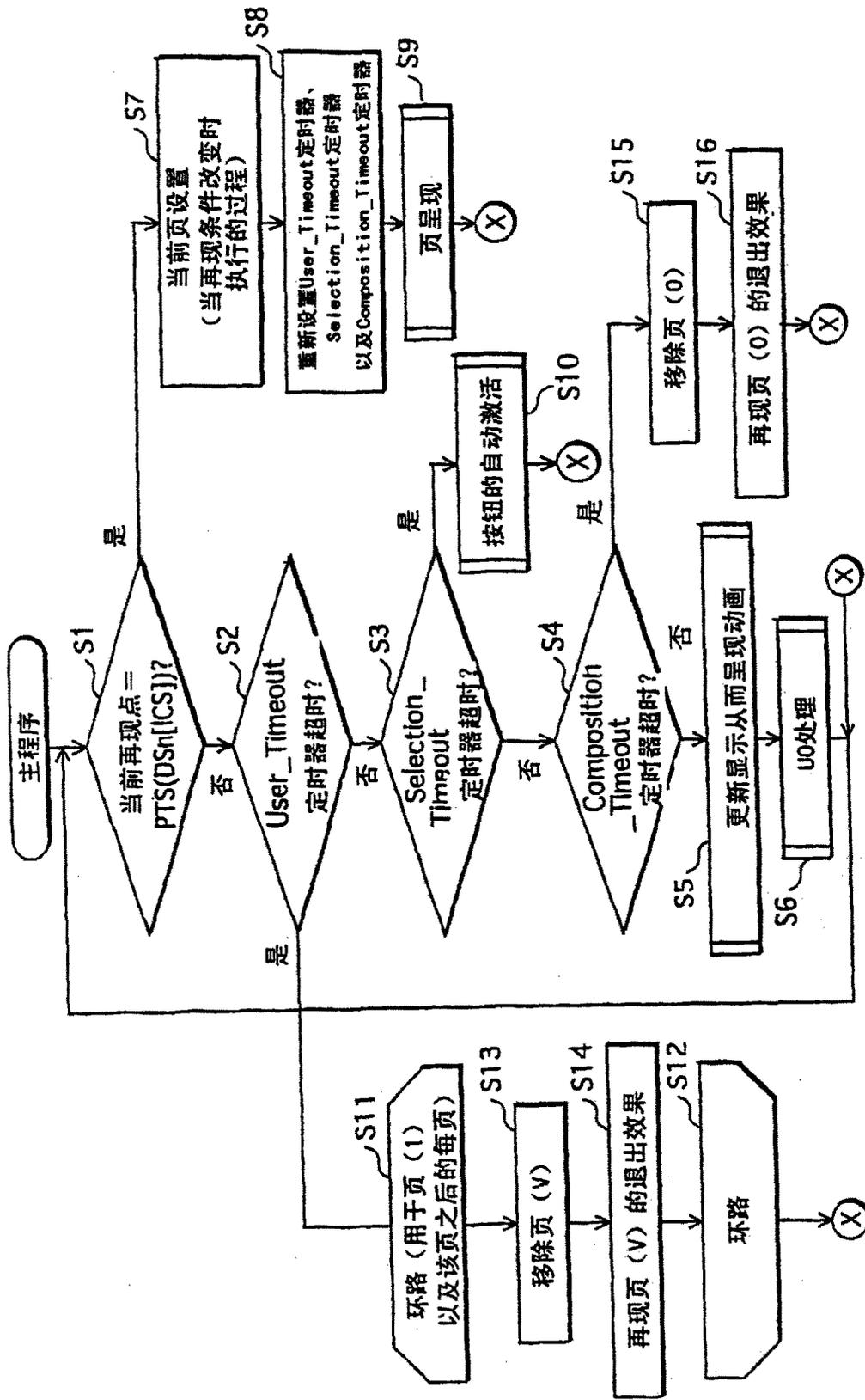


图 41

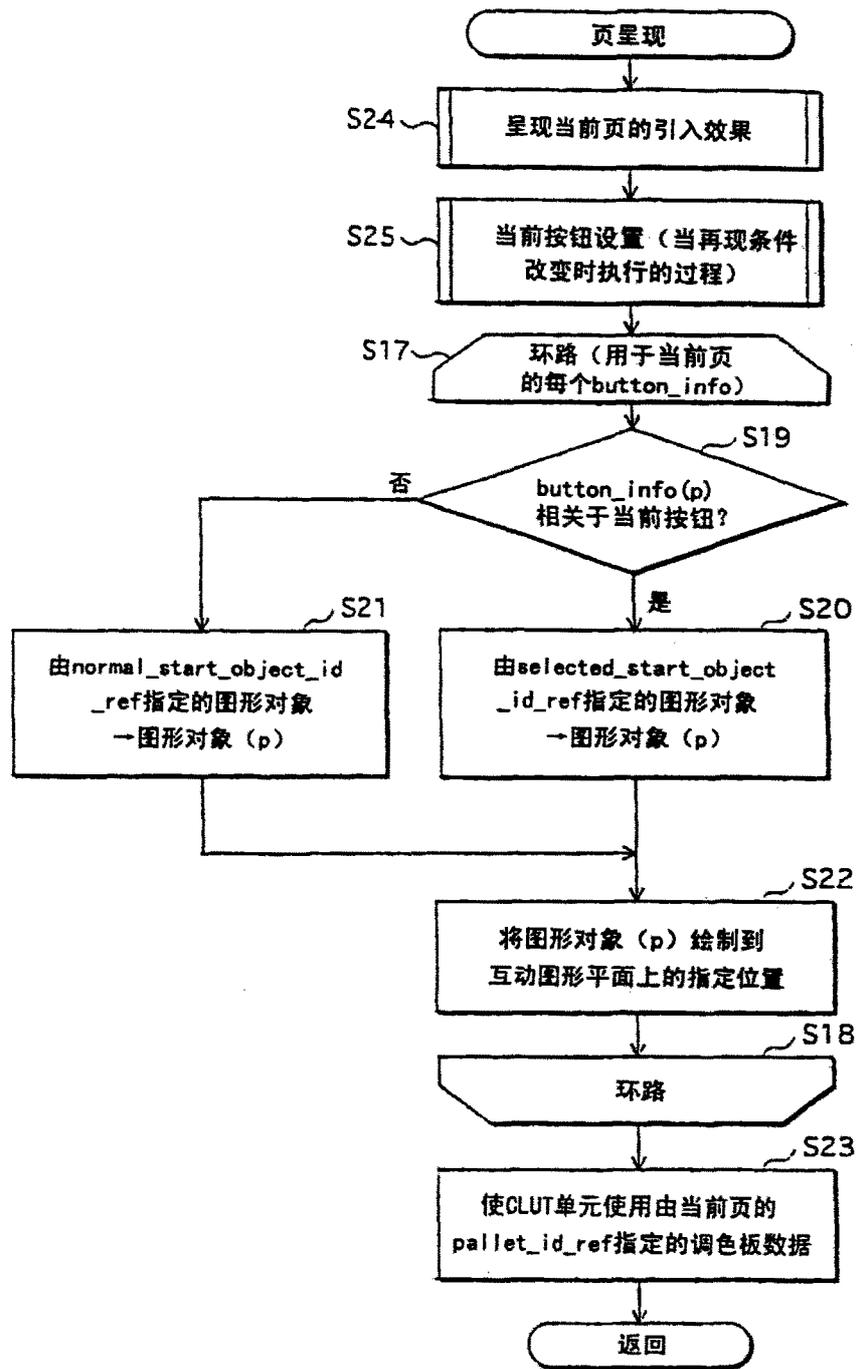


图 42

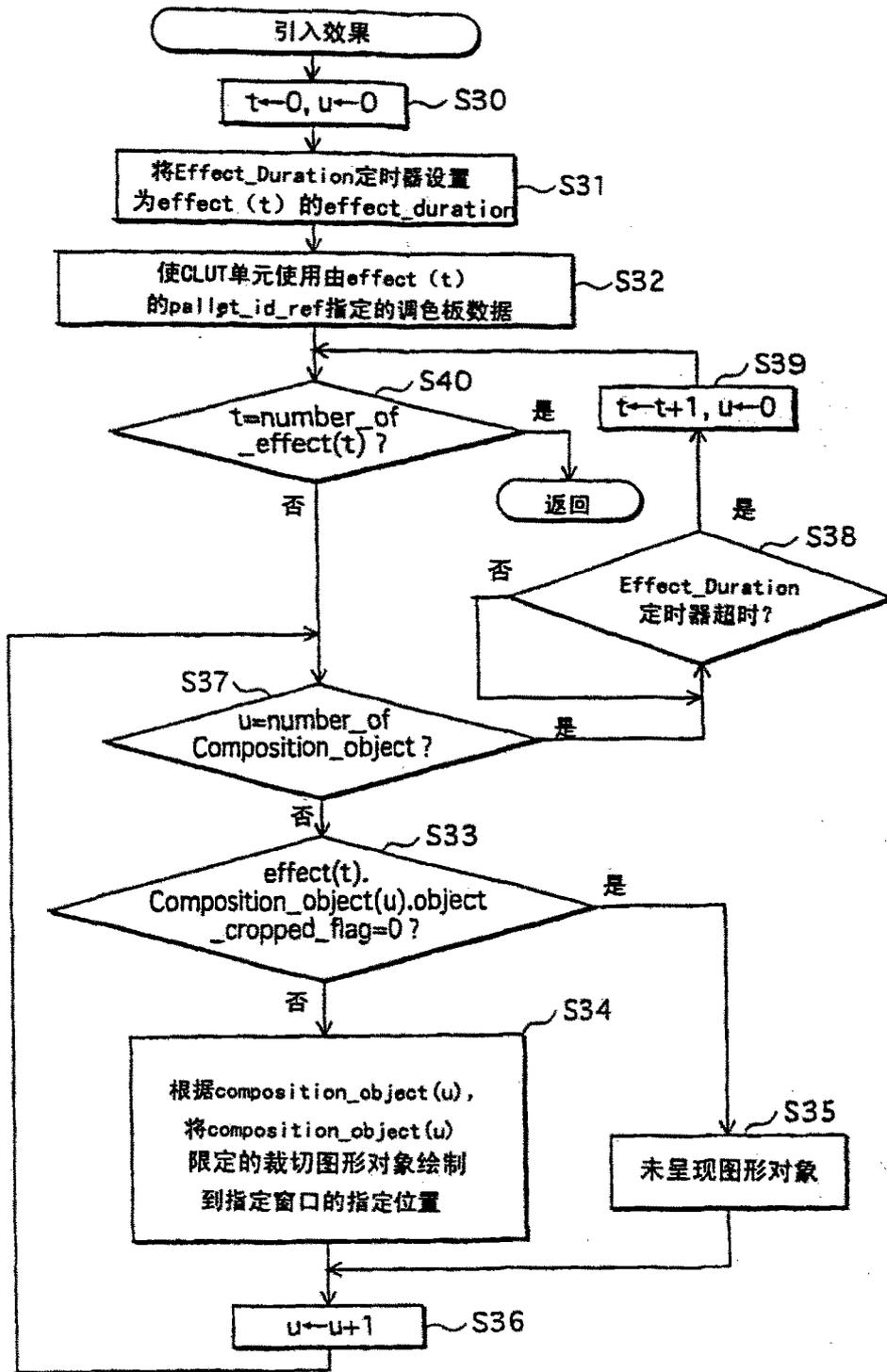


图 43

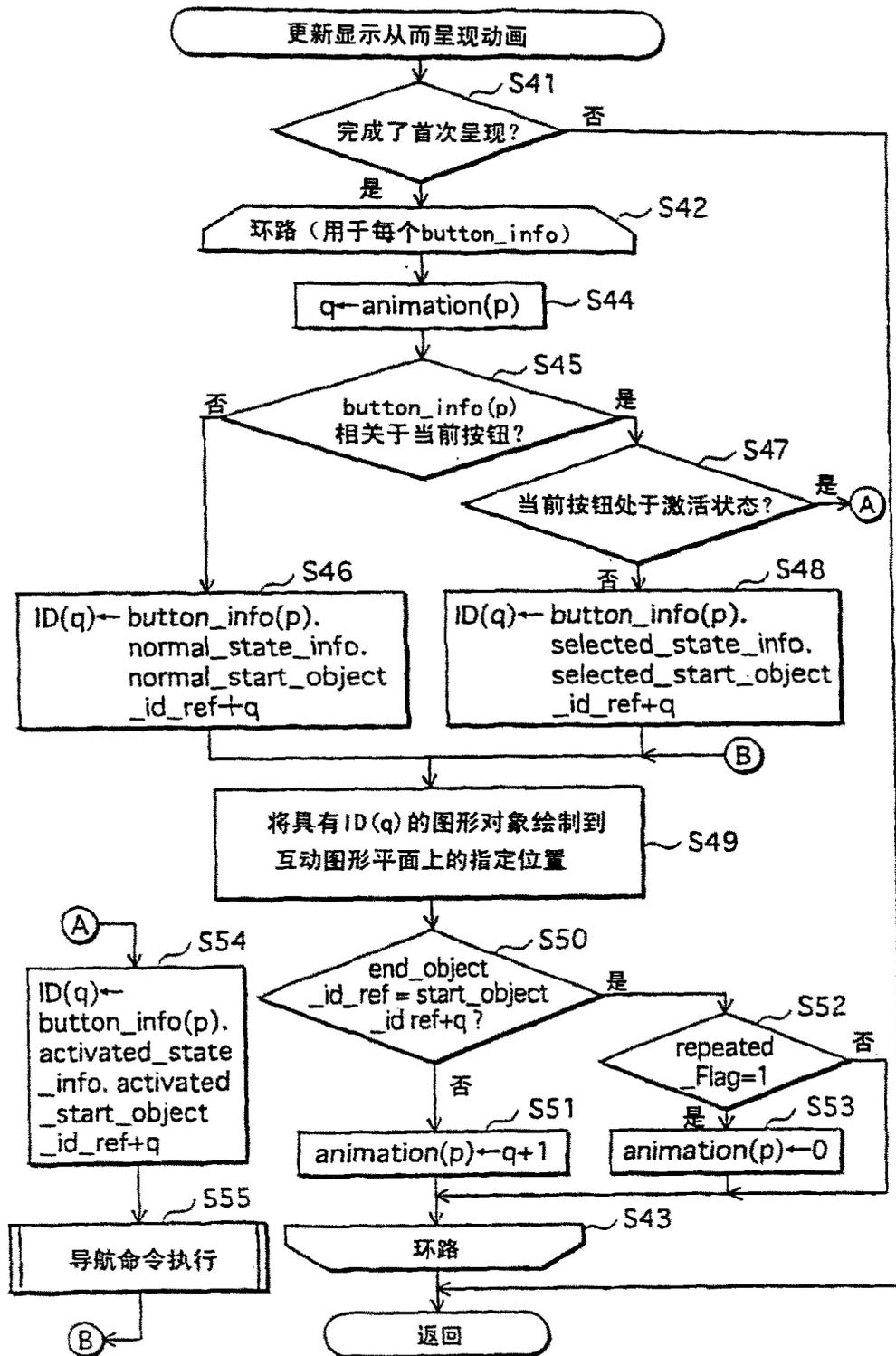


图 44

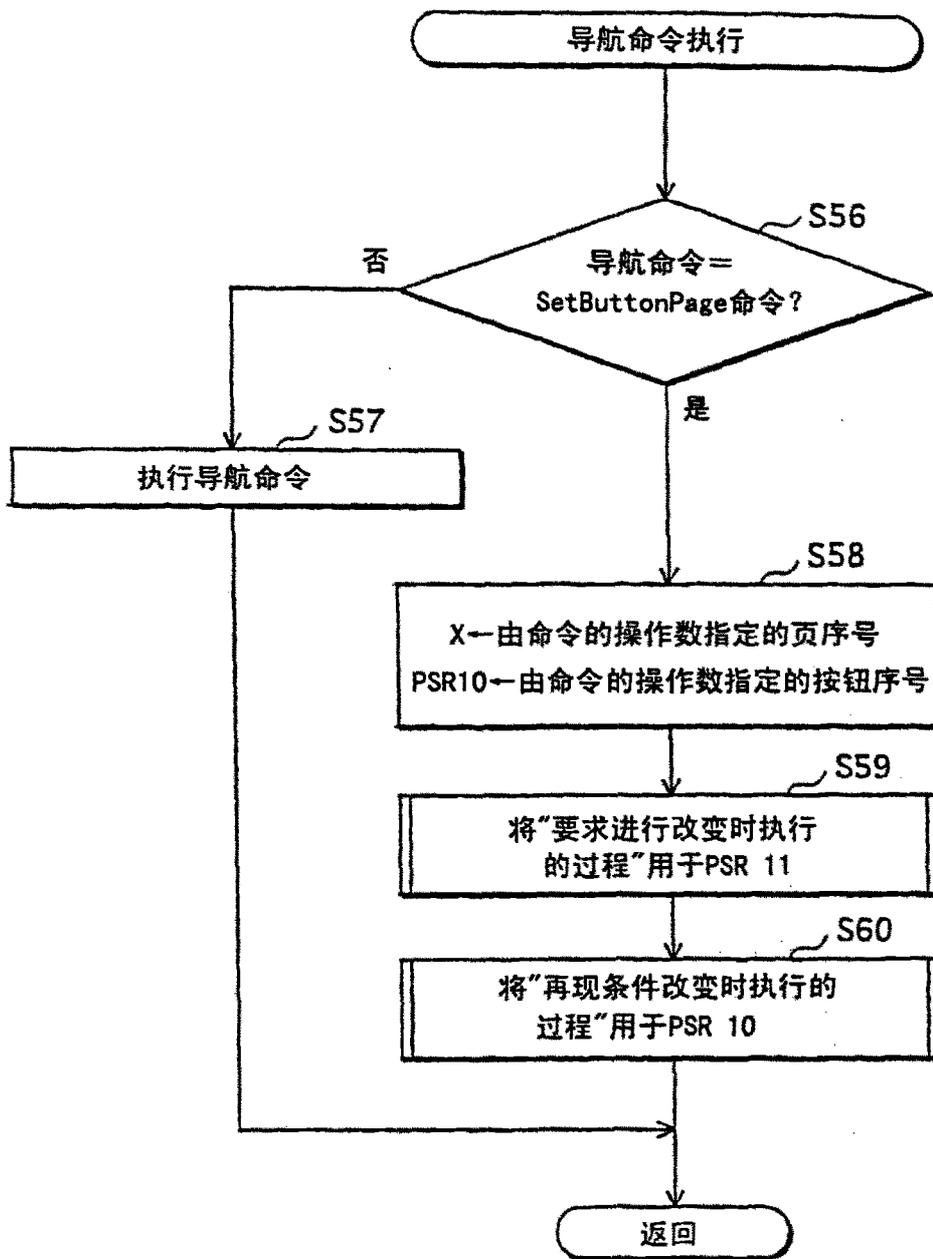


图 45

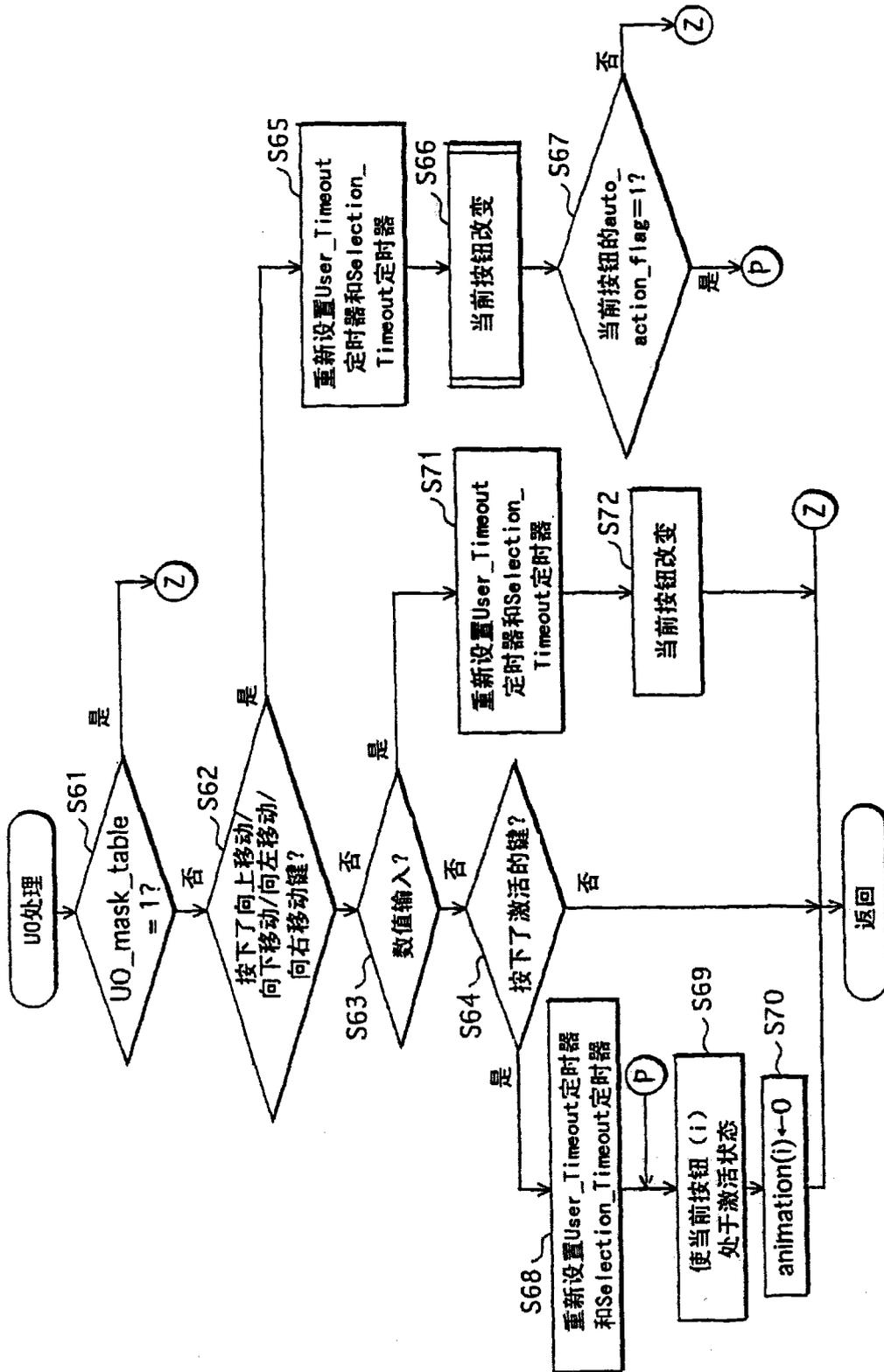


图 46

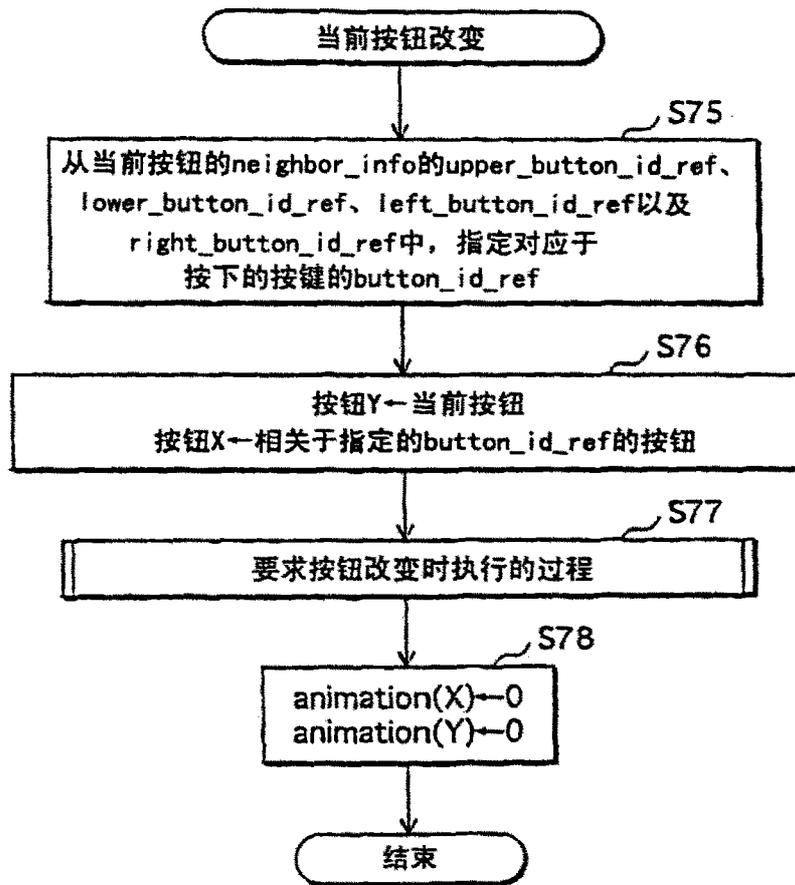


图 47

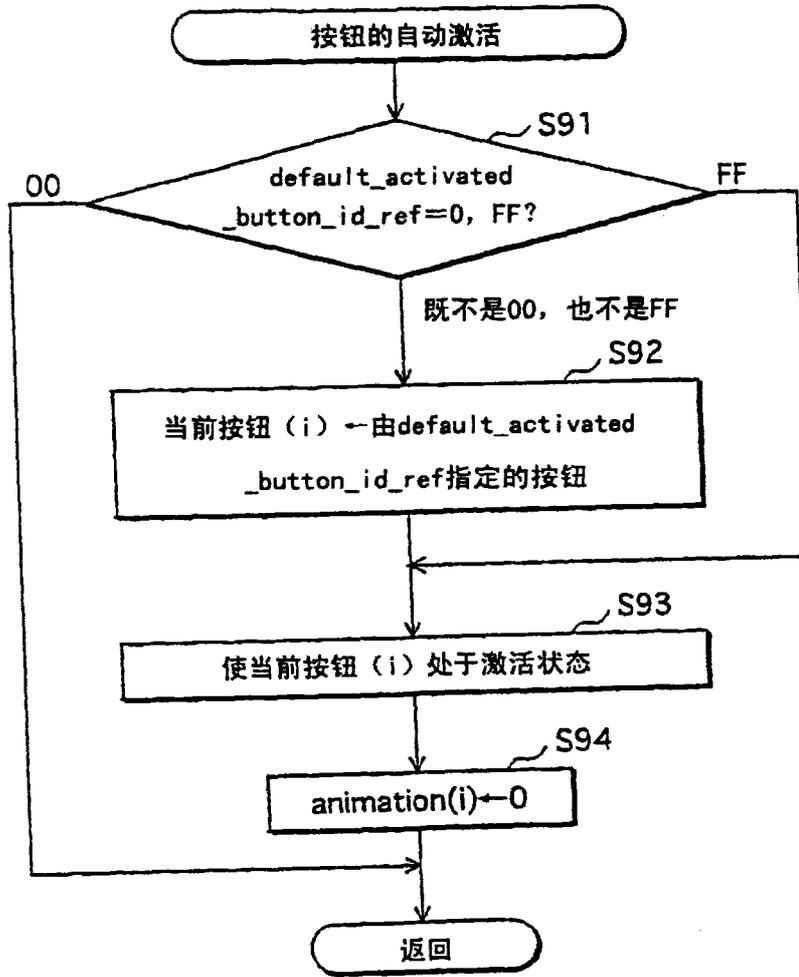


图 48

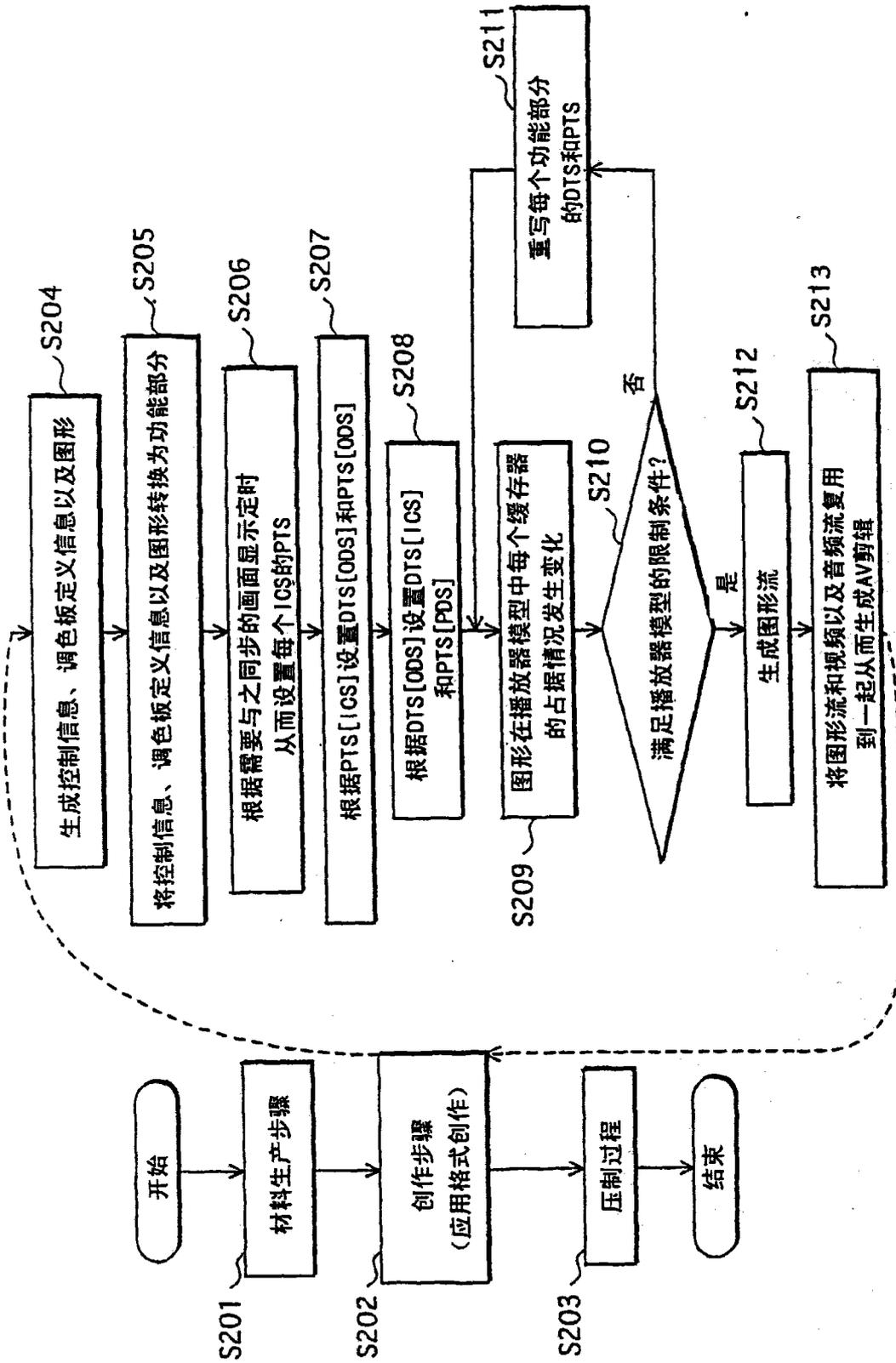


图 49