

(12) 发明专利

(10) 授权公告号 CN 101233749 B

(45) 授权公告日 2011. 11. 16

(21) 申请号 200680028386. 0

H04N 5/445(2006. 01)

(22) 申请日 2006. 08. 05

H04N 21/472(2011. 01)

(30) 优先权数据

- 10-2006-0073116 2006. 08. 02 KR
- 60/705, 491 2005. 08. 05 US
- 60/789, 577 2006. 04. 06 US
- 60/812, 090 2006. 06. 09 US

(56) 对比文件

- 同上.
- JP 2003271285 A, 2003. 09. 26, 全文.
- JP 9081358 A, 1997. 03. 28, 全文.
- US 6204887 B1, 2001. 03. 20, 说明书第 3 栏第 36 行至第 11 栏第 30 行、附图 1-5.
- JP 2004178127 A, 2004. 06. 24, 全文.
- 同上.
- US 20030142037 A1, 2003. 07. 31, 说明书第 2 页第 28 段至第 3 页第 31 段、第 4 页第 40-43 段、附图 2-3 和 5-6.
- US 6084553 A, 2000. 07. 04, 全文.

(85) PCT 申请进入国家阶段日
2008. 02. 01

(86) PCT 申请的申请数据
PCT/KR2006/003087 2006. 08. 05

(87) PCT 申请的公布数据
W02007/018388 EN 2007. 02. 15

(73) 专利权人 三星电子株式会社
地址 韩国京畿道

审查员 孙佳琛

(72) 发明人 李钟昊 李光基 郑云教
格莱恩·A·亚当斯

(74) 专利代理机构 北京铭硕知识产权代理有限公司 11286
代理人 郭鸿禧 刘奕晴

(51) Int. Cl.
H04N 5/45(2006. 01)

权利要求书 5 页 说明书 15 页 附图 26 页

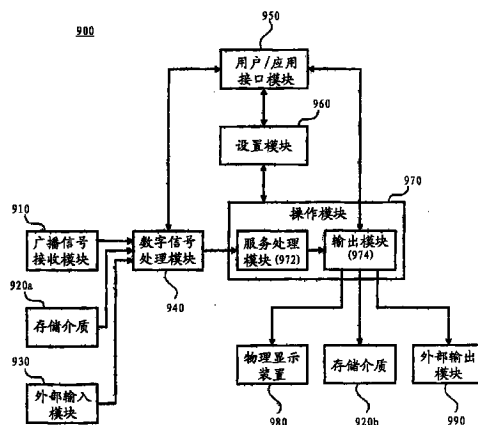
(54) 发明名称

用于提供多个屏幕的设备以及动态配置多个屏幕的方法

(57) 摘要

提供一种用于提供多个屏幕的设备以及动态配置多个屏幕的方法。所述用于提供多个屏幕的设备能够在单个物理显示装置上提供多个内容，并在存在于系统上的屏幕之间重新构建形式或连接关系。用于提供多个屏幕的设备包括：操作模块，产生至少一个用于显示服务的屏幕；以及设置模块，感测屏幕的设置的变化。

CN 101233749 B



1. 一种用于提供多个屏幕的设备,所述设备包括:

操作模块,产生用于显示服务的多个屏幕;以及

设置模块,感测屏幕的设置的变化,

其中,所述多个屏幕包括多个逻辑屏幕和多个显示屏幕,其中,所述多个逻辑屏幕被映射到所述多个显示屏幕,其中,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,服务被映射到该虚拟屏幕,

其中,操作模块将多个服务映射到所述多个逻辑屏幕,将所述多个逻辑屏幕布置在所述多个显示屏幕上,并经由多个输出端口提供所述多个显示屏幕,

其中,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

2. 如权利要求 1 所述的设备,其中,所述多个屏幕包括:逻辑屏幕,其被连接到包括在服务中的应用;以及显示屏幕,逻辑屏幕被映射到所述显示屏幕,其中,应用指示服务中的数据组件。

3. 如权利要求 2 所述的设备,其中,操作模块在显示屏幕上布置执行应用的逻辑屏幕。

4. 如权利要求 2 所述的设备,其中,设置的变化包括:执行应用的默认屏幕的改变、连接到屏幕的服务的语境的改变、显示屏幕的改变、在显示屏幕的区域中逻辑屏幕被映射的区域的坐标的改变、屏幕的视频端口的改变、屏幕的可视性的改变、以及显示逻辑屏幕的顺序的改变中的一个。

5. 如权利要求 1 所述的设备,其中,所述多个屏幕包括画中画屏幕。

6. 如权利要求 1 所述的设备,其中,所述多个屏幕包括画外画屏幕。

7. 一种用于提供多个屏幕的设备,所述设备包括:

操作模块,产生用于显示接收的服务的多个屏幕;以及

设置模块,改变屏幕的大小或坐标,

其中,所述多个屏幕包括多个逻辑屏幕和多个显示屏幕,其中,所述多个逻辑屏幕被映射到所述多个显示屏幕,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,所述接收的服务被映射到该虚拟屏幕,

其中,操作模块将多个服务映射到所述多个逻辑屏幕,将所述多个逻辑屏幕布置在所述多个显示屏幕上,并经由多个输出端口提供所述多个显示屏幕,

其中,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

8. 如权利要求 7 所述的设备,其中,所述多个屏幕包括画中画屏幕。

9. 如权利要求 7 所述的设备,其中,所述多个屏幕包括画外画屏幕。

10. 如权利要求 7 所述的设备,其中,所述多个屏幕包括:逻辑屏幕,其被连接到包括在所述多个服务中的应用;以及显示屏幕,逻辑屏幕被映射到所述显示屏幕,其中,应用指示服务中的数据组件。

11. 如权利要求 10 所述的设备,其中,操作模块在显示屏幕上布置执行应用的逻辑屏幕。

12. 如权利要求 10 所述的设备,其中,设置模块改变在显示屏幕的区域中逻辑屏幕被映射的区域的坐标以及逻辑屏幕的大小。

13. 一种用于提供多个屏幕的设备,所述设备包括:

服务处理模块,产生用于显示多个服务的多个逻辑屏幕以及所述多个逻辑屏幕被映射到的多个显示屏幕,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,所述多个服务中的每一个被映射到该虚拟屏幕;以及

设置模块,改变在所述多个显示屏幕上显示所述多个逻辑屏幕的顺序,

其中,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

14. 如权利要求 13 所述的设备,其中,所述多个逻辑屏幕包括画中画屏幕。

15. 如权利要求 13 所述的设备,其中,所述多个逻辑屏幕包括画外画屏幕。

16. 一种用于提供多个屏幕的设备,所述设备包括:

服务处理模块,产生用于显示多个服务的多个逻辑屏幕以及所述多个逻辑屏幕被映射到的多个显示屏幕,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,所述多个服务中的每一个被映射到该虚拟屏幕;以及

设置模块,根据多种布置方法中的一种布置方法在所述多个显示屏幕上布置所述多个逻辑屏幕,

其中,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

17. 如权利要求 16 所述的设备,其中,布置方法包括:

第一布置方法,在所述多个显示屏幕上显示所述多个服务;

第二布置方法,其中,根据第一布置方法的所述多个显示屏幕与至少一个重叠的屏幕一起存在;

第三布置方法,其中,显示服务的所述多个屏幕相应于与显示屏幕的整个区域映射的逻辑屏幕,显示服务的屏幕相应于与显示屏幕映射的至少一个未重叠的逻辑屏幕,显示服务的屏幕与重叠的屏幕不相应;

第四方法,其中,根据第三方法的所述多个屏幕与至少一个重叠的屏幕一起存在;

第五方法,其中,显示服务的所述多个屏幕相应于与显示屏幕的整个区域映射的至少两个未重叠的逻辑屏幕,但是与重叠的屏幕不相应;

第六方法,其中,根据第五方法的所述多个屏幕与至少一个重叠的屏幕一起存在;以及

第七方法,其中,显示服务的所述多个屏幕没有包括在第一到第六方法中。

18. 如权利要求 16 所述的设备,其中,所述多个逻辑屏幕包括画中画屏幕。

19. 如权利要求 16 所述的设备,其中,所述多个逻辑屏幕包括画外画屏幕。

20. 如权利要求 16 所述的设备,其中,设置模块参考包括在多个服务中的应用的权限范围来布置所述多个逻辑屏幕,其中,应用指示服务中的数据组件。

21. 如权利要求 16 所述的设备,其中,通过包括在服务中的应用或通过用户来选择布置方法,其中,应用指示服务中的数据组件。

22. 一种用于提供多个屏幕的设备,所述设备包括:

操作模块,产生用于显示多个服务的多个屏幕;以及

设置模块,限制在所述多个屏幕上执行包括在所述多个服务中的应用,

其中,所述多个屏幕包括多个逻辑屏幕和多个显示屏幕,其中,所述多个逻辑屏幕被映

射到所述多个显示屏幕,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,所述多个服务中的每一个被映射到该虚拟屏幕,

其中,操作模块将所述多个服务映射到所述多个逻辑屏幕,将所述多个逻辑屏幕布置在所述多个显示屏幕上,并经由多个输出端口提供所述多个显示屏幕,

其中,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个,应用指示服务中的数据组件。

23. 如权利要求 22 所述的设备,其中,所述多个屏幕包括画中画屏幕。

24. 如权利要求 22 所述的设备,其中,所述多个屏幕包括画外画屏幕。

25. 如权利要求 22 所述的设备,其中,不论用于指示应用可在所述多个屏幕上执行的信号是否包括在所述多个服务中,所述设置模块限制在所述多个屏幕上执行包括在所述多个服务中的应用。

26. 一种提供多个屏幕的方法,所述方法包括:

产生用于显示多个服务的多个屏幕,其中,所述多个屏幕包括多个逻辑屏幕和多个显示屏幕,其中,所述多个逻辑屏幕被映射到所述多个显示屏幕,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,所述多个服务被映射到该虚拟屏幕;

将多个服务映射到所述多个逻辑屏幕;

将所述多个逻辑屏幕布置在所述多个显示屏幕上;

经由多个输出端口提供所述多个显示屏幕;以及

感测所述多个屏幕的设置的变化,

其中,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

27. 如权利要求 26 所述的方法,其中,所述多个屏幕包括:逻辑屏幕,其被连接到包括在服务中的应用;以及显示屏幕,逻辑屏幕被映射到所述显示屏幕,其中,应用指示服务中的数据组件。

28. 如权利要求 27 所述的方法,还包括:在显示屏幕上布置执行应用的逻辑屏幕。

29. 如权利要求 27 所述的方法,其中,设置的变化包括:执行应用的默认屏幕的改变、连接到屏幕的服务的语境的改变、显示屏幕的改变、在显示屏幕的区域中逻辑屏幕被映射的区域的坐标的改变、屏幕的视频端口的改变、屏幕的可视性的改变、以及显示逻辑屏幕的顺序的改变中的一个。

30. 如权利要求 26 所述的方法,其中,所述多个屏幕包括画中画屏幕。

31. 如权利要求 26 所述的方法,其中,所述多个屏幕包括画外画屏幕。

32. 一种提供多个屏幕的方法,所述方法包括:

产生用于显示多个服务的多个屏幕,其中,所述多个屏幕包括多个逻辑屏幕和多个显示屏幕,其中,所述多个逻辑屏幕被映射到所述多个显示屏幕,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,所述多个服务中的每一个被映射到该虚拟屏幕;

将所述多个服务映射到所述多个逻辑屏幕;

将所述多个逻辑屏幕布置在所述多个显示屏幕上;

经由多个输出端口提供所述多个显示屏幕；以及
改变所述多个屏幕的大小或坐标，

其中，服务指示一组服务组件，服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

33. 如权利要求 32 所述的方法，其中，所述多个屏幕包括画中画屏幕。

34. 如权利要求 32 所述的方法，其中，所述多个屏幕包括画外画屏幕。

35. 如权利要求 32 所述的方法，其中，所述多个屏幕包括：逻辑屏幕，其被连接到包括在服务中的应用；以及显示屏幕，逻辑屏幕被映射到所述显示屏幕，其中，应用指示服务中的数据组件。

36. 如权利要求 35 所述的方法，还包括：在显示屏幕上布置执行应用的逻辑屏幕。

37. 如权利要求 35 所述的方法，其中，改变屏幕的大小或坐标的步骤包括：改变在显示屏幕的区域中逻辑屏幕被映射的区域的坐标以及逻辑屏幕的大小。

38. 一种提供多个屏幕的方法，所述方法包括：

产生用于显示多个服务的多个逻辑屏幕以及所述多个逻辑屏幕被映射到的多个显示屏幕，显示屏幕指示向外部装置实际输出的屏幕，逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕，所述多个服务中的每一个被映射到该虚拟屏幕；

将所述多个服务映射到所述多个逻辑屏幕；

将所述多个逻辑屏幕布置在所述多个显示屏幕上；

经由多个输出端口提供所述多个显示屏幕；以及

改变在所述多个显示屏幕上显示所述多个逻辑屏幕的顺序，

其中，服务指示一组服务组件，服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

39. 如权利要求 38 所述的方法，其中，所述多个逻辑屏幕包括画中画屏幕。

40. 如权利要求 38 所述的方法，其中，所述多个逻辑屏幕包括画外画屏幕。

41. 一种提供多个屏幕的方法，所述方法包括：

产生用于显示多个服务的多个逻辑屏幕以及所述多个逻辑屏幕被映射到的多个显示屏幕，显示屏幕指示向外部装置实际输出的屏幕，逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕，所述多个服务中的每一个被映射到该虚拟屏幕；以及

根据多种布置方法中的一种布置方法在所述多个显示屏幕上布置所述多个逻辑屏幕，

其中，服务指示一组服务组件，服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

42. 如权利要求 41 所述的方法，其中，布置方法包括：

第一布置方法，在所述多个显示屏幕上显示所述多个服务；

第二布置方法，其中，根据第一布置方法的所述多个显示屏幕与至少一个重叠的屏幕一起存在；

第三布置方法，其中，显示服务的所述多个屏幕相应于与显示屏幕的整个区域映射的逻辑屏幕，显示服务的屏幕相应于与显示屏幕映射的至少一个未重叠的逻辑屏幕，显示服务的屏幕与重叠的屏幕不相应；

第四方法，其中，根据第三方法的所述多个屏幕与至少一个重叠的屏幕一起存在；

第五方法,其中,显示服务的所述多个屏幕相应于与显示屏幕的整个区域映射的至少两个未重叠的逻辑屏幕,但是与重叠的屏幕不相应;

第六方法,其中,根据第五方法的所述多个屏幕与至少一个重叠的屏幕一起存在;以及第七方法,其中,显示服务的所述多个屏幕没有包括在第一到第六方法中。

43. 如权利要求 41 所述的方法,其中,所述多个逻辑屏幕包括画中画屏幕。

44. 如权利要求 41 所述的方法,其中,所述多个逻辑屏幕包括画外画屏幕。

45. 如权利要求 41 所述的方法,其中,参考包括在服务中的应用的权限范围来布置所述多个屏幕,其中,应用指示服务中的数据组件。

46. 如权利要求 41 所述的方法,其中,通过包括在服务中的应用或通过用户来选择布置方法,其中,应用指示服务中的数据组件。

47. 一种提供多个屏幕的方法,所述方法包括:

产生用于显示多个服务的多个屏幕;以及

限制在所述多个屏幕上执行包括在所述多个服务中的应用,

其中,所述多个屏幕包括多个逻辑屏幕和多个显示屏幕,所述多个逻辑屏幕被映射到所述多个显示屏幕,并被布置在所述多个显示屏幕上,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,所述多个服务中的每一个被映射到该虚拟屏幕,

其中,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个,应用指示服务中的数据组件。

48. 如权利要求 47 所述的方法,其中,所述多个屏幕包括画中画屏幕。

49. 如权利要求 47 所述的方法,其中,所述多个屏幕包括画外画屏幕。

50. 如权利要求 47 所述的方法,其中,不论用于指示应用可在所述多个屏幕上执行的信号是否包括在所述多个服务中,限制执行应用的步骤限制在所述多个屏幕上执行包括在所述多个服务中的应用。

51. 一种包括用于在物理显示装置上配置多个屏幕的设置模块的系统,其中,所述设置模块执行以下操作中的至少一个:(i) 改变每个屏幕的大小和坐标,(ii) 改变在物理显示装置上显示屏幕的顺序,(iii) 提取当前激活的屏幕的屏幕布置方法,(iv) 限制在屏幕上执行应用,

其中,所述多个屏幕包括多个逻辑屏幕和多个显示屏幕,其中,所述多个逻辑屏幕被映射到所述多个显示屏幕,并被布置在所述多个显示屏幕上,显示屏幕指示向外部装置实际输出的屏幕,逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,服务被映射到该虚拟屏幕,

其中,应用指示服务中的数据组件,服务指示一组服务组件,服务组件中的每一个包括视频组件、音频组件和数据组件中的至少一个。

52. 如权利要求 51 所述的系统,其中,设置模块感测多个屏幕中的改变。

用于提供多个屏幕的设备以及动态配置多个屏幕的方法

技术领域

[0001] 与本发明一致的设备和方法涉及提供多个屏幕并动态配置多个屏幕,更具体地说,涉及在单个物理显示装置上提供多个内容并在存在于系统上的屏幕之间重新构建形式或连接关系。

背景技术

[0002] 现有技术的广播接收机(诸如数字电视(TV)或数字机顶盒)在单个物理显示装置上仅提供一个内容元素,或者在单个物理显示装置上同时显示主屏幕和子屏幕。

[0003] 即使现有技术的广播接收机在同一显示屏幕上可同时显示主屏幕和子屏幕两者,但是它们仅能够按照有限数量的形式来排列主屏幕和子屏幕。在主屏幕之内显示的内容的情况下,内容的所有元素(即,视频数据、音频数据和其它数据)被显示。另一方面,在子屏幕之内显示的内容的情况下,只有内容的某些元素被显示。

[0004] 内容源包括广播服务(诸如卫星广播设备、地面广播设备或有线广播设备)、存储介质(诸如数字通用盘(DVD))或连接到输入端的外部装置。然而,很难使用现有的广播接收机在显示屏幕上显示由所述各种内容源提供的内容。

[0005] 在交互式TV应用程序环境(诸如多媒体家庭平台(MHP)、高级通用应用平台(ACAP)和开放式有线电视应用平台(OCAP))中,假设在物理显示装置上仅输出一个屏幕。

[0006] 在交互式TV应用程序环境中,例如,采用基于家庭音频/视频交互(HAVi)的用户接口(UI)。根据HAVi UI标准,即使对于在物理显示装置上显示的屏幕数量没有加以限制,但是通常只有一个屏幕显示在物理显示装置上。

发明内容

[0007] 技术问题

[0008] 在所述环境中,难以在独立的屏幕上显示多媒体内容的同时,针对显示在屏幕上的多媒体内容之一执行诸如解码、数字信号处理、用户交互处理等操作。此外,还难以动态控制应用程序的生命周期并按照屏幕为单位来使用资源。

[0009] 因此,需要一种在动态配置的屏幕上显示各种内容的方法。

[0010] 技术方案

[0011] 本发明提供一种用于动态配置多个屏幕的设备和方法,该设备和方法在单个物理显示装置上提供多个内容,从而在存在于系统上的屏幕之间重新构建形式或连接关系。

[0012] 根据本发明的一方面,提供一种用于提供多个屏幕的设备,所述设备包括:操作模块,产生至少一个用于显示服务的屏幕;以及设置模块,感测屏幕的设置的变化。

[0013] 根据本发明的另一方面,提供一种用于提供多个屏幕的设备,所述设备包括:操作模块,产生至少一个用于显示服务的屏幕;以及设置模块,改变屏幕的大小或坐标。

[0014] 根据本发明的另一方面,提供一种用于提供多个屏幕的设备,所述设备包括:服务处理模块,产生用于显示服务的逻辑屏幕以及逻辑屏幕被映射到的显示屏幕;以及设置模

块,改变在显示屏幕上显示逻辑屏幕的顺序。

[0015] 根据本发明的另一方面,提供一种用于提供多个屏幕的设备,所述设备包括:服务处理模块,产生用于显示服务的逻辑屏幕以及逻辑屏幕被映射到的显示屏幕;以及设置模块,根据一种或多种布置方法中的一种布置方法在显示屏幕上布置逻辑屏幕。

[0016] 根据本发明的另一方面,提供一种用于提供多个屏幕的设备,所述设备包括:操作模块,产生至少一个用于显示服务的屏幕;以及设置模块,限制在屏幕上执行包括在服务中的应用。

[0017] 根据本发明的另一方面,提供一种提供多个屏幕的方法,所述方法包括:产生至少一个用于显示服务的屏幕;以及感测屏幕的设置的变化。

[0018] 根据本发明的另一方面,提供一种提供多个屏幕的方法,所述方法包括:产生至少一个用于显示服务的屏幕;以及改变屏幕的大小或坐标。

[0019] 根据本发明的另一方面,提供一种提供多个屏幕的方法,所述方法包括:产生用于显示服务的逻辑屏幕以及逻辑屏幕被映射到的显示屏幕;以及改变在显示屏幕上显示逻辑屏幕的顺序。

[0020] 根据本发明的另一方面,提供一种提供多个屏幕的方法,所述方法包括:产生用于显示服务的逻辑屏幕以及逻辑屏幕被映射到的显示屏幕;以及根据一种或多种布置方法中的一种布置方法在显示屏幕上布置逻辑屏幕。

[0021] 根据本发明的另一方面,提供一种提供多个屏幕的方法,所述方法包括:产生至少一个用于显示服务的屏幕;以及限制在屏幕上执行包括在服务中的应用。

附图说明

[0022] 通过参照附图进行的对本发明示例性实施例的详细描述,本发明的上述和其它方面将会变得更加清楚,其中:

[0023] 图 1A 到图 1H 是示出根据本发明示例性实施例的 PiP 屏幕的配置的示图;

[0024] 图 2 是示出根据本发明示例性实施例的逻辑屏幕与显示屏幕之间的关系;的示图;

[0025] 图 3A 到图 3E 是示出根据本发明示例性实施例的包括映射器的屏幕的配置的示图;

[0026] 图 4 是示出根据本发明示例性实施例的服务源的框图;

[0027] 图 5A 和图 5B 是示出根据本发明示例性实施例的非抽象服务和抽象服务的示图;

[0028] 图 6 是示出逻辑屏幕和显示屏幕的属性信息和接口的示图;

[0029] 图 7 是示出根据本发明示例性实施例的逻辑屏幕的属性“z-order”的示图;

[0030] 图 8A 和图 8B 是均示出根据本发明示例性实施例的逻辑屏幕的属性“Display_Area”的示图;

[0031] 图 9 是示出根据本发明示例性实施例将两个服务映射到显示屏幕的方法的示图;

[0032] 图 10 是示出根据本发明示例性实施例的用于提供多个屏幕的设备的配置的框图;

[0033] 图 11 是示出根据本发明示例性实施例的动态配置多个屏幕的方法的流程图;

[0034] 图 12 是示出根据本发明示例性实施例的用于提供多个屏幕的软件架构的示图;

[0035] 图 13 是示出根据本发明示例性实施例的构成应用编程接口 (API) 层的模块之间

的关系的示图；

[0036] 图 14 是示出根据本发明示例性实施例的由图 13 所示的模块将显示在各个相应的逻辑屏幕上的多个服务显示在显示屏幕上的方法的流程图；

[0037] 图 15 是示出根据本发明示例性实施例的设置模块的操作的状态图；

[0038] 图 16A 到图 16E 是示出根据本发明示例性实施例的用于设置屏幕的代码的示例的示图；以及

[0039] 图 17 是示出根据本发明示例性实施例的由图 13 所示的模块来交换将被显示在逻辑屏幕上的服务的处理的流程图。

具体实施方式

[0040] 通过参照以下对示例性实施例的详细描述和附图，可更加容易地理解本发明的优点和特点以及实现本发明的方法。然而，本发明可按照许多不同的形式来实现，并且不应被解释为受限于这里阐述的示例性实施例。而是提供这些示例性实施例，从而所述公开将是彻底和完整的，并将本发明的构思完全传达给本领域的技术人员，并且本发明将仅由权利要求来限定。在整个说明书中，相同的标号指示相同的部件。

[0041] 以下将参照根据本发明实施例的用户接口、方法和计算机程序产品的流程图来描述本发明。应理解：流程图中的每个块以及流程图中的块的组合可通过计算机程序指令来实现。可将这些计算机程序指令提供给通用计算机、专用计算机或其它可编程数据处理设备的处理器以产生机器，从而经由计算机或其它可编程数据处理设备的处理器执行的指令创建用于实现在流程图的一个或多个块中指定的功能的手段。

[0042] 这些计算机程序指令还可被存储在计算机可用或计算机可读存储器中，其能够指导计算机或其它可编程数据处理设备按照特定方式来工作，从而存储在计算机可用或计算机可读存储器中的指令产生制造产品，其包括实现在流程图的一个或多个块中指定的功能的指令手段。

[0043] 计算机程序指令还可被加载到计算机或其它可编程数据处理设备，以促使将在计算机或其它可编程设备上执行的一系列操作步骤产生计算机实现的处理，从而在计算机或其它可编程设备上执行的指令提供用于实现在流程图的一个或多个块中指定的功能。

[0044] 流程图中的每个块代表模块、段或部分代码，其包括一个或多个用于实现指定的逻辑功能的可执行指令。应注意到：在某些可替换实现中，在块中标注的功能可不按顺序发生。例如，连续示出的两个块实际上可基本同时执行，或者所述块有时候可按照相反的顺序来执行，这取决于涉及的功能。

[0045] 为了更好地理解本发明，现将定义在该公开中使用的术语。

[0046] 术语“服务”指示一组一起显示的多媒体内容，即，一组服务组件 (component)。

[0047] 服务组件是服务的元素，并包括视频组件、音频组件和数据组件。数据组件是服务中的应用。

[0048] 此外，可将应用分类为非绑定应用和服务绑定应用。由于非绑定应用具有高优先级，所以资源被平稳地分配。监视器应用相应于具有最高优先级的特定非绑定应用。

[0049] 服务绑定应用与传输流相关，并允许低于非绑定应用的优先级。服务绑定应用不用作关键系统 (critical system)。当出现对于资源的竞争时，服务绑定应用比非绑定应用

更有可能放弃资源分配。服务绑定应用包括结合类型和非结合类型,所述结合类型按照与正被传输的流相结合的方式操作,所述非结合类型依赖于所述流来操作。

[0050] 术语“服务语境”指示可控制服务的执行的对象,并包括提供服务所需的各种资源、装置和执行状态信息。

[0051] 术语“物理显示装置”指示实际显示服务的内容的物理空间。

[0052] 术语“显示屏幕”指示在物理显示装置上实际显示的屏幕。可在显示屏幕中直接设置任意服务,并且可在物理显示装置上显示所述显示屏幕。或者,可在物理显示装置上显示至少一个被映射到显示屏幕的特定区域的逻辑屏幕。

[0053] 术语“逻辑屏幕”指示任意服务被显示的空间。逻辑屏幕是被映射到显示屏幕之前的虚拟屏幕,因此不被显示在物理显示装置上。

[0054] 逻辑屏幕和显示屏幕可以是背景静止图像、视频光栅和图形光栅的组合、图形光栅可以是文本、线条、色彩和图像的组合或视频帧的混合。

[0055] 术语“主服务”指示由用户通过显示在物理显示装置上的菜单或遥控器,或者由应用通过 API 选择作为主要服务的服务,显示主服务的屏幕被称为“主屏幕”。

[0056] 术语“画中画服务”(PiP 服务)指示由用户通过显示在物理显示装置上的菜单或遥控器,或者由应用经由 API 选择作为主服务中的子服务的服务,PiP 服务可被显示在画中画屏幕(PiP 屏幕)或主屏幕上。

[0057] PiP 屏幕包括如图 1A 到图 1D 所示的占据另一屏幕的一部分的屏幕,以及如图 1E 到图 1F 所示的在不与另一屏幕重叠的情况下与所述另一屏幕同时显示的屏幕。在这种情况下,应理解:PiP 屏幕可包括如图 1G 和图 1H 所示的在物理显示装置中的任意位置或区域上显示的或与另一屏幕重叠的屏幕。

[0058] 图 2 是示出根据本发明示例性实施例的逻辑屏幕与显示屏幕之间的关系关系的示图。

[0059] 参照图 2,可使用逻辑屏幕 210、212 和 214 来显示服务。可通过映射块 230 将逻辑屏幕 210、212 和 214 映射到显示屏幕 220、222 和 224。

[0060] 具体说来,逻辑屏幕 210 和 212 被映射到显示屏幕 220,逻辑屏幕 212 和 214 被映射到逻辑屏幕 222,逻辑屏幕 210、212 和 214 被映射到显示屏幕 224。

[0061] 简言之,至少一个显示服务的逻辑屏幕被映射到显示屏幕的任意区域。

[0062] 映射块 230 是用于将逻辑屏幕映射到显示屏幕所需的多条信息的组。所述多条信息的示例包括:多个逻辑屏幕中的每一个被映射到的显示屏幕上的预定区域的坐标信息、所述逻辑屏幕和显示屏幕的标识信息、以及指定按照什么顺序将逻辑屏幕显示在显示屏幕上的信息。

[0063] 映射块 230 可改变逻辑屏幕的大小,以便在显示屏幕的任意区域中进行分配。也就是说,映射块 230 可执行逻辑屏幕的缩放及其位置的分配,图 3A 到图 3E 是示出包括作为映射块的映射器(mapper)的屏幕的配置的示图。

[0064] 参照图 3A,由映射器按照正常大小将主屏幕映射到整个显示屏幕,其中,所述主屏幕包括背景静止图像 B、视频光栅 V 和图形光栅 G 的组合。由映射器按照减小的大小将仅包括视频组件的 PiP 屏幕映射到整个显示屏幕。在这种情况下,将映射的 PiP 屏幕显示在主屏幕上,这根据 Z 值来确定。参考字符 Z 是指稍后将描述的 z-order 值。可将覆盖屏幕(overlay screen)与显示屏幕组合。覆盖屏幕是布置在最外侧的特定屏幕,并且可在提供

字幕功能时使用所述覆盖屏幕。如图 3A 所示, PiP 屏幕可仅具有视频组件, 或如图 3B 所示, PiP 屏幕可具有背景静止图像 B、视频光栅 V 和图形光栅 G 的组合。

[0065] 参照图 3C, 由映射器按照正常大小将主屏幕映射到整个显示屏幕, 其中, 主屏幕包括背景静止图像 B、视频光栅 V 和图形光栅 G 的组合。由映射器按照减小的大小将仅具有视频组件的两个 PiP 屏幕 #1 和 #2 映射到显示屏幕的任意区域。在这种情况下, 映射的 PiP 屏幕被布置在主屏幕上, 并且可恒定地保持 Z 值。此外, 可将覆盖屏幕与显示屏幕组合。屏幕的配置可如图 3C 所示具有多个仅包括视频组件的 PiP 屏幕, 或者可如图 3D 所示具有多个包括背景静止图像 B、视频光栅 V 和图形光栅 G 的组的 PiP 屏幕。

[0066] 在图 3E 中示出画外画 (POP) 屏幕。可理解: 传统 PiP 屏幕显示在主屏幕内侧, POP 屏幕显示在主屏幕外侧。参照图 3E, 由映射器按照减小的大小将多个 PiP 屏幕 #1 和 #2 映射到显示屏幕的任意区域, 其中, 所述多个 PiP 屏幕 #1 和 #2 包括背景静止图像 B、视频光栅 V 和图形光栅 G 的组合。在这种情况下, 可恒定地保持映射的 PiP 屏幕 #1 和 #2 的 Z 值。此外, 可将覆盖屏幕与显示屏幕组合。

[0067] 可通过由将被执行的各种计算机编程语言准备的接口或功能来实现映射块 230, 并且映射块 230 可通过将上述信息用作参数来创建或改变逻辑屏幕与显示屏幕之间的关系。

[0068] 此外, 可通过具有逻辑屏幕与显示屏幕之间的映射功能的硬件来实现映射块 230。

[0069] 此外, 如图 4 所示, 可将由各种服务源提供的服务显示在显示屏幕上, 并且可将显示屏幕显示在物理显示装置上。

[0070] 存在提供广播服务的服务源 (诸如地面广播设备 320 和有线广播设备 330)、提供存储在存储介质中的服务的服务源 (诸如个人视频录像机 (PVR) 340)、以及经由有线网络或无线网络提供服务的服务源 (图 4 中未示出)。

[0071] 广播接收机 310 从服务源接收服务, 并产生显示每个服务的逻辑屏幕。

[0072] 然后, 使用预定义的方法或使用由用户或应用设置的方法, 在将被显示在物理显示装置的显示屏幕上直接设置任意服务。否则, 在物理显示装置 350 上显示至少一个被映射到显示屏幕上的任意区域的逻辑屏幕。简言之, 在物理显示装置 350 上显示由地面广播设备 320、有线广播设备 330 和 PVR 提供的服务。

[0073] 地面广播设备 320、有线广播设备 330 和 PVR340 在图 4 中被显示为服务源, 但是本发明并不受限于此。根据本发明的示例性实施例, 提供可被一起显示的多媒体内容的任何类型的多媒体内容源均可成为服务源。

[0074] 如图 5A 和图 5B 所示, 根据本发明示例性实施例的服务可被分类为抽象服务和非抽象服务。

[0075] 抽象服务不是由实时发送的广播信号提供的服务, 而是独立于广播频道的服务。抽象服务仅包括数据组件, 即, 应用, 而不包括视频组件和音频组件。抽象服务的示例包括具有基于 OCAP 标准的非绑定应用的服务。

[0076] 将非抽象服务理解为除了抽象服务之外的服务。

[0077] 根据本发明的当前示例性实施例, 抽象服务与非抽象服务两者均具有独立性。例如, 可不通过逻辑屏幕而在物理显示装置上直接设置抽象服务, 并且可在逻辑屏幕上显示非抽象服务。然后, 可将逻辑屏幕映射到设置了抽象服务的显示屏幕。然后, 可通过物理显

示装置来输出显示屏幕。通过这种处理,可独立于非抽象服务而将抽象服务显示在显示屏幕上。此外,可将抽象服务和非抽象服务映射到不同的逻辑屏幕。然后,可将逻辑屏幕映射到单个显示屏幕。换言之,可独立于非抽象服务将抽象服务显示在显示屏幕上。

[0078] 根据本发明的当前示例性实施例,可将逻辑屏幕和显示屏幕分类为不同的对象。或者,屏幕可根据一个屏幕对象的属性信息而用作逻辑屏幕或显示屏幕。

[0079] 具体说来,屏幕对象是逻辑屏幕还是显示屏幕取决于屏幕对象属性中的“Type”信息。

[0080] 屏幕对象属性包括:“Type”、“z-Order”、“Display_Area”、“Visibility”、“Associated_Display_Screen”、“Associated_Service_Contexts”和“OutputPort”。

[0081] 图 6 示出屏幕对象的属性信息以及用于处理屏幕的属性的接口。

[0082] 属性“Type”510 用于确定屏幕类型 - 逻辑屏幕或显示屏幕。

[0083] 属性“z-Order”520 用于确定按照什么顺序将沿 z 轴排列多个逻辑屏幕。图 7 示出对于逻辑屏幕的属性“z-Order”的值的组合,物理显示装置上的逻辑屏幕的配置。

[0084] 参照图 7,第一逻辑屏幕 620 和第二逻辑屏幕 630 被分别映射到显示屏幕 610 的预定区域。具体说来,第一逻辑屏幕 620 被显示在显示屏幕 610 上,第二逻辑屏幕 630 与第一逻辑屏幕 620 部分重叠地显示在所述显示屏幕上。换言之,沿着 z 轴的方向顺序地排列显示屏幕 610、第一逻辑屏幕 620 和第二逻辑屏幕 630。在这种情况下,第一逻辑屏幕 620 的属性“z-Order”可被设置为值 1,第二逻辑屏幕 630 的属性“z-Order”可被设置为值 2。可将第一逻辑屏幕 620 和第二逻辑屏幕 630 的属性“z-Order”设置为任何数字或字符,只要它们能够代表沿着 z 轴排列第一逻辑屏幕 620 和第二逻辑屏幕 630 的特定顺序。

[0085] 如将在图 8A 和图 8B 中所示,属性“Display_Area”530 是关于逻辑屏幕的显示屏幕区域的信息。

[0086] 图 8A 示出逻辑屏幕 710 被映射到显示屏幕 720 的整个区域,图 8B 示出逻辑屏幕 730 被映射到显示屏幕 740 的部分区域。

[0087] 属性“Display_Area”可包括指定逻辑屏幕被映射到的显示屏幕的预定部分的二维坐标,或者可包括指定显示屏幕上的预定位置和偏移值的信息,所述偏移值指示逻辑屏幕偏离显示屏幕上的预定位置的程度。

[0088] 属性“Visibility”540 确定逻辑屏幕将以可见还是不可见的方式显示在显示屏幕上。可通过变更属性“Visibility”530 的值使得逻辑屏幕出现在显示屏幕上或从显示屏幕消失。

[0089] 属性“Associated-Display_Screen”550 是关于与逻辑屏幕相关的显示屏幕的信息。与任何显示屏幕均不相关的逻辑屏幕不会被显示在物理显示装置上,也不会被发送到外部输出装置。

[0090] 属性“Associated_Service_Contexts”560 是关于连接到逻辑屏幕或显示屏幕的服务语境的信息。在所述服务语境中设置的服务可被显示在逻辑屏幕或显示屏幕上。

[0091] 属性“OutputPort”570 是关于输出显示屏幕的装置的信息,所述装置包括:显示屏幕、有线/无线通信介质和各种存储介质。

[0092] 可提供用于识别或变更图 6 所示的属性的值的接口。参照图 6,所述接口可包括:接口“SET”,用于设置属性值或将逻辑屏幕连接到显示屏幕;接口“ADD”,用于添加属性值

或将逻辑屏幕连接到服务；接口“GET”，用于识别属性值；以及“REMOVE”，用于删除属性值。这些接口可包括分别执行相应功能的处理、函数、进程或方法。

[0093] 例如，方法“getDisplayScreen(void)”返回与当前屏幕相关的显示屏幕。具体说来，如果当前屏幕是逻辑屏幕，则所述方法“getDisplayScreen(void)”返回相关的显示屏幕。如果当前屏幕是显示屏幕，则方法“getDisplayScreen(void)”返回关于当前屏幕的参考信息。此外，如果当前屏幕是逻辑屏幕，但是不存在相关屏幕，则方法“getDisplayScreen(void)”返回值“NULL”。

[0094] 根据另一示例，方法“public void setDisplayArea(HScreenRectangle rect) throws SecurityException, IllegalStateException”提供用于将当前逻辑屏幕映射到相关显示屏幕的预定区域的函数。被作为参数提供的实例属于包“org.havi.ui”的类“HScreenRectangle”，并具有二维位置信息。可进行方法“SecurityException”和“IllegalStateException”的执行，作为方法“setOutputScreen(HScreen screen)”的异常操作。在当前屏幕是逻辑屏幕时，或者当与当前逻辑屏幕相关的显示屏幕的一部分由于主机平台的特征而无法改变时，可执行方法“IllegalStateException”。

[0095] 根据另一示例，方法“getOutputArea(void)”返回当前屏幕的区域信息，作为HScreenRectangle信息。如果当前屏幕相应于显示屏幕，则方法“getOutputArea(void)”返回与HScreenRectangle(0,0,1,1)具有相同值的HScreenRectangle信息。如果当前屏幕是逻辑屏幕，则方法“getOutputArea(void)”返回关于由当前屏幕占据的显示屏幕上的区域的信息。如果当前屏幕是逻辑屏幕但是不与任何显示屏幕相关，则方法“getOutputArea(void)”返回值“NULL”。

[0096] 特定术语在下面的描述中用于指示特定接口。然而，本领域的技术人员将认识到：对特定功能命名仅仅是为了指示它的功能性。对示例性实施例的所述详细描述并不是为了区分名称有区别但是功能没有区别的功能。

[0097] 图9是示出在将被映射到单个显示屏幕的两个逻辑屏幕上设置两个服务的处理的示图。

[0098] 参照图9，第一服务包括全部三种服务组件，即，视频、音频和数据组件，第二服务仅包括视频和音频组件。然而，本发明并不对服务组件加以任何限制，图9所示的第一和第二服务是示例性的。

[0099] 如图9所示，按照与现有技术几乎相同的方式在物理显示装置上显示第一和第二服务。根据本发明的当前示例性实施例，在不对可显示在单个显示屏幕上的服务的数量加以任何限制的情况下，可彼此独立地在物理显示装置上显示多个服务。

[0100] 图10是根据本发明示例性实施例的用于提供多个屏幕的设备的框图。

[0101] 参照图10，用于提供多个屏幕的设备900包括：数字信号处理模块940、操作模块970、用户/应用接口模块950和设置模块960。

[0102] 此外，设备900包括：广播信号接收模块910、存储介质920a和作为服务源的外部输入模块930，并包括：物理显示装置970、存储介质980、存储介质920b和作为服务输出介质的外部输出模块990。

[0103] 如这里使用的术语“模块”表示但不受限于执行特定任务的软件或硬件组件（诸如现场可编程门阵列（FPGA）或专用集成电路（ASIC）、模块可被方便地配置为驻留在可寻

址存储介质上,并且可被配置为在一个或多个处理器上执行。因此,作为示例,模块可包括:组件(诸如软件组件、面向对象的软件组件、类组件和任务组件)、处理、函数、属性、进程、子程序、程序代码段、驱动程序、固件、微代码、电路、数据、数据库、数据结构、表、数组和变量。在组件和模块中提供的功能可被组合为较少的组件和模块,或者被进一步分离为附加的组件和模块。

[0104] 数字信号处理模块 940 从广播信号接收模块 910、存储介质 920a 或外部输入模块 930 接收服务(诸如多媒体内容)的各种信息,例如,视频信息、音频信息或数据信息。

[0105] 广播信号接收模块 910 接收卫星、地面或有线广播信号,并发送接收的广播信号,存储介质 920a 存储服务的视频信息、音频信息或数据信息,外部输入模块 930 从外部装置(诸如连接到网络的网络接口模块)接收服务的视频信息、音频信息或数据信息。

[0106] 数字信号处理模块 940 使用接收的服务组件来恢复多个服务。恢复的服务包括抽象或非抽象服务。

[0107] 这里,短语“多个服务”是指由广播信号接收模块 910 发送的两个或多个服务,或者由广播信号接收模块 910 和存储介质 920a 分别发送的两个或多个服务。

[0108] 数字信号处理模块 940 可借助于用户/应用接口模块 950 根据用户或应用的选择来恢复服务。在这种情况下,用户或应用可选择任意服务与屏幕之间的连接。

[0109] 操作模块 970 显示由数字信号处理模块 940 恢复的服务,然后将逻辑屏幕映射到显示屏幕或直接映射到显示屏幕上的服务。为此,操作模块 940 可包括服务处理模块 972 或输出模块 974。

[0110] 服务处理模块 972 产生一个或多个逻辑屏幕,用来显示由数字信号处理模块 940 恢复的服务。

[0111] 输出模块 974 将由服务处理模块 972 产生的多个逻辑屏幕映射到显示屏幕。可借助于用户/应用接口模块 950 使用预定义的方法或由用户设置的方法来进行逻辑屏幕到显示屏幕的映射。

[0112] 由数字信号处理模块 940 恢复的服务可以不通过服务处理模块 972 来处理。相反,由数字信号处理模块 972 恢复的服务可以被直接映射到由输出模块 974 产生的显示屏幕的特定部分。

[0113] 由输出模块 974 提供的显示屏幕可被显示在物理显示装置 980 上,或者可被存储在存储介质 920b 中。存储介质 920b 的示例包括:计算机可读软盘、硬盘、CD-ROM、DVD、DVD-ROM、BD(蓝光盘)和半导体存储器。

[0114] 此外,由输出模块 974 提供的显示屏幕可以被发送到经由外部输出模块 990 连接到网络的外部装置。

[0115] 为此,输出模块 974 可包括多个输出端口,经由所述输出端口,可提供显示屏幕。在这种情况下,可借助于用户/应用接口模块 950,经由预先设置为默认的输出端口或由用户选择的输出端口来提供显示屏幕。

[0116] 设置模块 960 设置存在于系统上的屏幕的形式及其连接关系。这里,术语“设置”表示重新构建产生的显示屏幕的形式及其连接关系。为了设置所述形式和连接关系,设置模块 960 可感测屏幕的设置的变化。

[0117] 这里,屏幕包括显示服务的逻辑屏幕以及与所述逻辑屏幕相映射的显示屏幕。设

置的改变包括以下改变中的任何一个：执行应用的默认屏幕的改变、连接到屏幕的服务的语境的改变、显示屏幕的改变、在显示屏幕的区域中逻辑屏幕被映射的区域的坐标的改变、屏幕的视频端口的改变、屏幕的可视性的改变或显示逻辑屏幕的顺序的改变。

[0118] 设置模块 960 执行设置,以便改变屏幕的大小或坐标(以下,称为“第一设置”),改变在显示屏幕上显示逻辑屏幕的顺序(以下,称为“第二设置”),改变布置屏幕的方法(以下,称为“第三设置”),限制屏幕上应用的执行(以下,称为“第四设置”)。

[0119] 当第一设置被执行时,设置模块 960 可改变显示屏幕区域中逻辑屏幕被映射的区域的坐标以及逻辑屏幕的大小。

[0120] 当第三设置被执行时,设置模块 960 可根据至少一个布置方法在显示屏幕区域上布置逻辑屏幕。

[0121] 布置方法包括:第一布置方法,在显示屏幕上显示服务;第二布置方法,其中,根据第一布置方法的显示屏幕与至少一个重叠的屏幕一起存在;第三布置方法,其中,显示服务的屏幕相应于与显示屏幕的整个区域映射的逻辑屏幕,显示服务的屏幕相应于与显示屏幕映射的至少一个未重叠的逻辑屏幕,显示服务的屏幕与重叠的屏幕不相应;第四方法,其中,根据第三方法的屏幕与至少一个重叠的屏幕一起存在;第五方法,其中,显示服务的屏幕相应于与显示屏幕的整个区域映射的至少两个未重叠的逻辑屏幕,但是与重叠的屏幕不相应;第六方法,其中,根据第五方法的屏幕与至少一个重叠的屏幕一起存在;以及第七方法,其中,显示服务的屏幕没有包括在上述任何一种方法中。

[0122] 设置模块 960 可参考包括在服务中的应用的权限范围来布置屏幕。也就是说,只有当用于改变布置方法的权限被给予应用时,设置模块 960 可改变屏幕布置方法。可通过提供相应服务的服务提供者将所述权限给予应用。

[0123] 当第四设置被执行时,设置模块 960 可忽略包括在服务中的信号。也就是说,即使指示可在屏幕上执行应用的信号被包括在服务中,设置模块 960 也可限制屏幕上的应用的执行。

[0124] 另一方面,根据从服务提供者提供的应用或用户控制命令来执行设置模块 960。

[0125] 用户或应用可使用用户/应用接口模块 950 来选择多个服务之一或恢复期望的服务。此外,用户可使用用户/应用接口模块 950 来选择多个显示屏幕之一。为了执行所述选择,用户/应用接口模块 950 可使用设置模块 960。

[0126] 由于图 10 所示的模块根据它们的功能来划分,所以能够被连接到其它模块。

[0127] 图 11 是示出根据本发明示例性实施例的动态配置多个屏幕的方法的流程图。

[0128] 通常,按照预定的格式(例如,MPEG 流格式)来发送构成多媒体内容的视频信息、音频信息和数据信息。在操作 S1010,用于提供服务(诸如多媒体内容服务)的设备接收视频信息、音频信息和数据信息,并基于所述视频信息、音频信息和数据信息来恢复服务。这里,可通过用户或应用来选择或预先确定在操作 S1010 恢复的服务。用户可使用显示在显示装置上的菜单或遥控器来选择任意屏幕与某一屏幕之间的连接。应用可使用 API 来选择连接。

[0129] 此外,数据信息包括关于对服务的应用的的应用信息,这些应用信息包括指示是否可在 PiP 屏幕上执行所述应用的信号信息。应用信息的示例包括基于 MHP 标准的应用信息表(AIT)和基于 OCAP 标准的 eXtended 应用信息表(XAIT)。所述信号信息可被添加到应用

信息。

[0130] 然后,在操作 S1020,恢复的服务被设置,从而其可显示在逻辑屏幕上。在操作 S1030,逻辑屏幕被映射到显示屏幕。在操作 S1040,使用显示屏幕、存储介质或网络将显示屏幕提供给用户。

[0131] 如上所述,在操作 S1050,当逻辑屏幕和显示屏幕被产生时,当逻辑屏幕与显示屏幕映射时,或者当显示屏幕被显示时,设置模块 960 设置显示的屏幕。由设置模块 960 执行的设置包括上述第一设置到第四设置,并且可根据通过用户 / 应用接口模块 950 输入的用户控制命令以及管理设置模块 960 的操作的应用的控制命令被执行。由于已经参照图 10 描述了设置,所以将省略对它的描述。

[0132] 恢复的服务在图 11 中被示为经由逻辑屏幕而显示在物理显示装置上。然而,恢复的服务可在不经过逻辑屏幕的情况下被直接显示在物理显示装置上。

[0133] 当用户选择 PiP 服务时,在两种模式下实现 PiP 服务。在第一模式下,只有在主屏幕上选择的 PiP 服务的视频组件被提供,而不创建 PiP 服务的单独逻辑屏幕,即,PiP 屏幕。在第二模式下,创建 PiP 服务的单独逻辑屏幕,以提供在创建的 PiP 屏幕上选择的 PiP 服务。

[0134] 为了简化,图 11 示出仅将一个服务映射到显示屏幕的方法。然而,多个服务可在经过或者不经过多个逻辑屏幕的情况下被映射到显示屏幕。

[0135] 当显示屏幕按照这种方式被提供给用户时,用户可执行多个服务。

[0136] 图 12 是示出根据本发明示例性实施例的用于提供多个屏幕的软件架构的示图。

[0137] 参照图 12,软件架构 1100 包括:装置驱动程序层 1110、API 层 1120 和应用层 1130。

[0138] 装置驱动程序层 1110 从各种多媒体内容源接收服务组件,并对接收的服务组件进行解码。接收的服务组件的示例包括:视频信息、音频信息和数据信息。

[0139] API 层 1120 产生逻辑屏幕和显示屏幕,并将服务、逻辑屏幕和显示屏幕彼此映射。

[0140] 应用层 1130 提供用户接口,从而用户可动态配置显示服务的逻辑屏幕,或者应用层 1130 将用户命令发送到 API 层 1120,从而 API 层 1120 可执行用户命令。

[0141] 为了执行上述操作,应用层 1130 可通过当前传送的应用或先前存储的应用来进行操作,可通过应用层 1130 来执行屏幕检索操作。

[0142] 用户使得装置驱动程序层 1110 能够借助于应用层 1130 来经由物理显示装置提供显示屏幕或将显示屏幕存储在存储介质 920b 中。此外,用户可使得装置驱动程序层 1110 能够将显示屏幕经由网络发送到外部装置。

[0143] 为此,装置驱动程序层 1110 可包括多个输出端口,其能够提供显示屏幕。否则,API 层 1120 可包括多个输出端口。

[0144] 为了在显示屏幕上动态地配置多个逻辑屏幕,如图 13 所示,API 层 1120 可包括多个软件模块,例如,多屏幕管理器模块“MultiScreenManager”1210、多屏幕语境模块“MultiScreenContext”1230、多屏幕语境监听器(contextlistener)模块“MultiScreenContextListener”1250 和多屏幕语境事件模块“MultiScreenContextEvent”1240。

[0145] 多屏幕管理器模块 1210 管理多屏幕语境模块 1230,检索期望的屏幕,显示指定什么装置被屏幕共享的信息,注册多屏幕语境监听器模块 1250 或取消多屏幕语境监听器模

块 1250 的注册。

[0146] 多屏幕语境模块 1230 是与屏幕对象 1220 相关的接口对象,并根据由多屏幕语境模块 1230 执行的接口操作来确定屏幕对象 1220 将成为逻辑屏幕还是显示屏幕。在多屏幕语境模块 1230 中可设置诸如图 6 所示的属性 510 到 570 的各种属性。多屏幕语境模块 1230 可提供以上参照图 6 描述的功能“SET”、“ADD”、“GET”和“REMOVE”。

[0147] 当通过多屏幕语境模块 1230 变更屏幕对象 1220 的属性信息时,多屏幕语境事件模块 1240 用作事件类(其宣告屏幕对象 1220 的属性信息已经改变),并且多屏幕语境监听器模块 1250 用作监听器接口对象,其可在预定的应用类中实现,所述预定的应用类尝试接收由多屏幕语境事件模块 1240 提示的事件。

[0148] 应用 1260 是在应用层 1130 上驱动模块。应用 1260 允许用户选择期望的服务并在显示屏幕上自由地排列多个逻辑屏幕。

[0149] 具体说来,应用 1260 将允许用户动态地配置和管理逻辑屏幕的各种命令发送到多屏幕管理器模块 1210,多屏幕管理器模块 1210 控制与各个命令相应的操作通过多屏幕语境模块 1230 被执行。

[0150] 多屏幕语境模块 1230 与屏幕对象 1220 相关,并管理图 6 所示的屏幕对象 1220 的属性信息。为了管理屏幕对象 1220 的属性信息,多屏幕语境模块 1230 可包括各种功能或方法。

[0151] 多屏幕管理器模块 1210 可从装置驱动程序层 1110 接收由各种服务源提供的服务组件,并执行用于在逻辑屏幕或显示屏幕上显示接收的服务组件的操作。

[0152] 图 14 是示出根据本发明示例性实施例的由图 13 所示的模块在显示屏幕上显示在各个相应的逻辑屏幕上显示的多个服务的方法的流程图。

[0153] 参照图 14,在操作 S1310,多屏幕管理器模块 1210 产生显示屏幕以及与将被执行的服务的数量相应的多个逻辑屏幕。

[0154] 在操作 S1320,多屏幕管理器模块 1210 将逻辑屏幕连接到从装置驱动程序层 1110 接收的各个相应服务。多屏幕管理器模块 1210 可通过将接收的服务的服务语境对象设置为逻辑屏幕服务的参数,对于逻辑屏幕中的每一个,调用方法“addServiceContext”。方法“addServiceContext”将逻辑屏幕连接到服务,并可通过多屏幕语境模块 1230 来提供所述方法“addServiceContext”。

[0155] 在操作 S1330,一旦逻辑屏幕被连接到各个服务,则多屏幕管理器模块 1210 将逻辑屏幕连接到显示屏幕。此时,多屏幕管理器模块 1210 可通过将逻辑屏幕被连接到的显示屏幕对象设置为参数,对于逻辑屏幕中的每一个,调用方法“setDisplayScreen”。方法“setDisplayScreen”将逻辑屏幕连接到显示屏幕,并可通过多屏幕语境模块 1230 来提供所述方法“setDisplayScreen”。

[0156] 可将方法“setDisplayScreen”设置到“public void setDisplayScreen(HScreen screen) throws SecurityException, IllegalStateException”,并且所述方法允许被作为参数提供的实例“HScreen”与当前逻辑屏幕相关。在这种情况下,实例“HScreen”优选的是显示屏幕。

[0157] 方法“setDisplayScreen(HScreen screen)”的参数可包括值“NULL”。在这种情况下,当在没有异常处理的情况下执行方法“setDisplayScreen(HScreen screen)”时,当前

逻辑屏幕不再与显示屏幕相关。

[0158] 可进行方法“SecurityException”和“IllegalStateException”的执行,作为对方法“setOutputScreen(HScreen screen)”的异常操作。

[0159] 在当前屏幕是逻辑屏幕时,或者当与当前逻辑屏幕相关的显示屏幕的一部分由于主机平台的特征而无法改变时,可执行方法“IllegalStateException”。

[0160] 在操作 S1340,确定逻辑屏幕将被分别映射到的显示屏幕上的区域。此时,由多屏幕语境模块 1230 提供的预定方法可被调用,以确定逻辑屏幕将被显示的显示屏幕上的区域。

[0161] 图 15 是示出根据本发明示例性实施例的设置模块 960 的操作的状态图。设置模块 960 可执行第一设置操作 1510 到第四设置操作 1540。

[0162] 为了执行上述设置操作,设置模块 960 可感测屏幕的设置的变化。

[0163] 为了感测设置的变化,设置模块 960 可通过使用方法“notify”来感测通知的事件。当应用的“HScreen”或“MultiScreenContext”的设置被改变并且包括参数“MultiScreenContextEvent”时,方法“notify”被调用。因此,应用应该注册监听器,以便感测方法“notify”的事件。

[0164] 为了注册或终止监听器,设置模块 960 可调用方法“addScreenContextListener”和方法“removeScreenContextListener”。方法“addScreenContextListener”和方法“removeScreenContextListener”被用于注册或终止监听器,并包括实例“MultiScreenContextListener”作为参数。

[0165] 方法“notify”的参数“MultiScreenContextEvent”是继承“java.util.EventObject”并用于提供如下的各种改变的类。也就是说,应用的基本 HScreen 的改变、连接的服务语境的改变、连接的输出 HScreen 的改变、在连接的输出 HScreen 中指定的区域的改变、连接的视频输出端口 VideoOutputPorts 的改变、屏幕的可视性的改变、屏幕的 z-order 的改变。

[0166] 为了感测上述改变,可使用方法“getId”。方法“getId”具有整数数据类型并返回标识符,所述标识符诸如 SCREEN_Z_ORDER_CHANGED、SCREEN_CONTEXT_CHANGED、SCREEN_DISPLAY_AREA_CHANGED、SCREEN_VISIBILITY_CHANGED、SCREEN_OUTPUT_PORT_CHANGED、SCREEN_CHANGED 和 SCREEN_DISPLAY_SCREEN_CHANGED。

[0167] 图 16A 是示出用于感测屏幕设置的变化代码的示例的示图。在图 16A 中示出接收设置的改变的类“AContextListener”被定义,定义的类的实例被产生,所述实例被注册为监听器、所述监听器被终止。

[0168] 为了执行第一设置操作 1510,设置模块 960 可调用方法“setDisplayArea”。方法“setDisplayArea”用于改变屏幕的大小或坐标,并可由类“MultiScreenConfigurableContext”来提供。

[0169] 方法“setDisplayArea”可被表示为“void setDisplayArea(org.havi.ui.HScreenRectangle rect) throws java.lang.SecurityException, java.lang.IllegalStateException”的形式,并且逻辑 HScreen 被连接到输出 HScreen 的区域。参数“rect”表示实例“HScreenRectangle”,其确定与由逻辑 HScreen 获得的区域相应的输出 HScreen 的区域。

[0170] 此外,当没有给予权限时,可执行异常“java.lang.SecurityException”。此外,当 HScreen 的形式与“SCREEN_TYPE_LOGICAL”不相应时,当连接到逻辑 HScreen 的输出 HScreen 的区域无法被改变时,或者当系统永久地使用连接的输出 HScreen 的预定区域时,可执行异常“java.lang.IllegalStateException”。

[0171] 图 16B 是示出用于改变屏幕的大小或坐标的代码以及当改变屏幕的大小或坐标时使用的代码的示例的示图。也就是说,通过应用给定类“MultiScreenConfigurableContext”的实例“configurableScreen”的方法“setDisplayArea”来改变屏幕的大小和坐标。

[0172] 为了执行第二设置操作 1520,设置模块 960 可调用方法“setZOrder”方法。“setZOrder”用于改变逻辑屏幕的 z-order,并且可从类“MultiScreenConfigurableContext”来提供。可通过“void setZOrder(int order) throws java.lang.SecurityException, java.lang.IllegalStateException”以及相同输出 HScreen 的逻辑 HScreen 中相应逻辑 HScreen 的 z-order 来表示方法“setZOrder”。

[0173] 此外,当没有给予权限时,可执行异常“java.lang.SecurityException”。当 HScreen 的形式与“SCREEN_TYPE_LOGICAL”不相应时,当相应逻辑 HScreen 的 z-order 无法被改变时,或当系统对于相应逻辑屏幕永久地使用预定 z-order 时,可执行异常“java.lang.IllegalStateException”。

[0174] 图 16C 是示出用于改变 z-order 的代码的示例的示图。也就是说,在图 16C 中,通过应用给定类“MultiScreenConfigurableContext”的实例“configurableScreen”的方法“setZOrder”,相应逻辑 HScreen 的 z-order 被改变为 3。

[0175] 为了执行第三设置操作 1530,设置模块 960 可调用方法“getScreenConfigurations”、方法“getScreenConfiguration”和方法“setScreenConfiguration”。

[0176] 方法“getScreenConfigurations”具有返回数据类型“MultiScreenConfiguration[]”,并且当提取从多屏幕提供装置 900 提供的屏幕布置方法时,使用所述方法“getScreenConfigurations”。

[0177] 方法“getScreenConfiguration”具有数据类型“MultiScreenConfiguration”,并且当提取当前激活的屏幕的屏幕布置方法时,使用所述方法“getScreenConfiguration”。

[0178] 当没有将权限给予方法“getScreenConfigurations”和方法“getScreenConfiguration”时,可执行异常“java.lang.SecurityException”。

[0179] 当激活选择的屏幕布置方法时,使用方法“setScreenConfiguration”。

[0180] 可通过“public void setScreenConfiguration(MultiScreenConfiguration configuration) throws java.lang.SecurityException, java.lang.IllegalArgumentException, java.lang.IllegalStateException”的形式来表示方法“setScreenConfiguration”。参数“configuration”表示“MultiScreenConfiguration”的实例,其指示将被激活的屏幕布置方法。

[0181] 当在方法“setScreenConfiguration”中没有给予权限时,可执行异常“java.lang.SecurityException”。

[0182] 可通过类“MultiScreenConfiguration”来提供方法

“getScreenConfigurations”、方法“getScreenConfiguration”和方法“setScreenConfiguration”。由类“MultiScreenConfiguration”提供的屏幕布置方法包括以下形式中的任何一个：单个屏幕的形式，其中，执行应用的屏幕相应于显示屏幕；重叠的屏幕的形式，其中，单个屏幕与至少一个重叠的屏幕一起存在；PiP 屏幕的形式，其中，执行应用的屏幕相应于映射到显示屏幕的整个区域的逻辑屏幕，相应于映射到显示屏幕的至少一个未重叠的逻辑屏幕，但是与重叠的屏幕不相应；重叠的 PiP 屏幕的形式，其中，PiP 屏幕与至少一个重叠的屏幕一起存在；PoP 屏幕的形式，其中，执行应用的屏幕相应于映射到显示屏幕的整个区域的至少两个未重叠的逻辑屏幕组合，但是与重叠的屏幕不相应；重叠的 PoP 屏幕的形式，其中，PoP 屏幕与至少一个重叠的屏幕一起存在；以及没有包括在上述任何一种形式中的形式。

[0183] 此外，为了检查相应屏幕的详细设置形式，搜索模块可调用方法“getScreenConfigurationType”。这里，方法“getScreenConfigurationType”用于返回当前激活的屏幕的设置形式并可由模块“MultiScreenConfiguration”来提供。

[0184] 例如，可通过以下形式来表示方法“getScreenConfigurationType”：“intgetScreenConfigurationType()”

[0185] {return SCREEN_CONFIGURATION_NON_PIP,

[0186] SCREEN_CONFIGURATION_NON_PIP_WITH_OVERLAY,

[0187] SCREEN_CONFIGURATION_PIP,

[0188] SCREEN_CONFIGURATION_PIP_WITH_OVERLAY,

[0189] SCREEN_CONFIGURATION_POP,

[0190] SCREEN_CONFIGURATION_POP_WITH_OVERLAY,

[0191] SCREEN_CONFIGURATION_GENERAL}”。方法“getScreenConfigurationType”以整数数据类型的形式返回当前激活的屏幕布置方法。这里，上述设置形式相应于单个屏幕、重叠的屏幕、PiP 屏幕、重叠的 PiP 屏幕、PoP 屏幕、重叠的 PoP 屏幕和一般屏幕。

[0192] 图 16D 是示出改变显示服务的屏幕的代码的示例的示图。在检查当前激活的屏幕的形式是 PiP 还是 PoP 之后，基于检查结果来改变设置形式。

[0193] 此外，为了执行第三设置操作 1530，设置模块 960 可调用方法“getScreens”。

[0194] 方法“getScreens”用于返回可访问屏幕的集合并具有包“org.havi.ui”的类“HScreen”的数据类型。也就是说，方法“getScreens”返回可访问实例“HScreen”的集合。实例“HScreen”的集合可包括由方法“org.havi.ui.HScreen.getDefaultHScreen”返回的实例以及由方法“org.havi.ui.HScreen.getHScreens”返回的实例。

[0195] 如上所述，将被返回的实例“HScreen”包括可访问调用的应用的各种屏幕实例。当基于 OCAP 标准的装置独立地调用应用时，实例“HScreen”的集合可包括能够访问基于 OCAP 标准的装置的各种屏幕实例。

[0196] 图 16E 是示出用于改变屏幕布置方法的代码的示例的示图。图 16E 示出通过由应用在预定 PoP 屏幕布置方法中选择的方法来布置屏幕。

[0197] 图 17 是示出由图 13 所示的模块来交换将被显示在逻辑屏幕上的服务的处理的流程图。

[0198] 在操作 S1410，多屏幕管理器模块 1210 临时终止将在两个逻辑屏幕之间交换的服

务。

[0199] 然后,在操作 S1420,在所述两个逻辑屏幕之间交换关于服务的信息。此时,对于所述两个逻辑屏幕中的每一个调用方法“removeServiceContext”和方法“addServiceContext”,从而在所述两个逻辑屏幕之间交换在所述两个逻辑屏幕中设置的服务语境。方法“removeContext”去除连接到逻辑屏幕的服务语境,方法“addServiceContext”将新的服务语境添加到逻辑屏幕。可通过多屏幕语境模块 1230 来提供方法“removeContext”和“addServiceContext”。根据本发明的当前示例性实施例,在逻辑服务之间交换关于连接到逻辑服务的服务的信息服务信息,从而获得交换主屏幕和子屏幕的效果。此外,根据本发明的当前示例性实施例,即使在显示屏幕上执行三个或多个服务,它们也能够任意数量的逻辑屏幕之间进行交换。

[0200] 在操作 S1430,产生服务与两个逻辑屏幕之间的连接已经被改变的事件的发生。然后,在操作 S1440,多屏幕语境事件模块 1240 将事件发送到多屏幕语境监听器模块 1250,新近连接到所述两个逻辑屏幕的服务开始被执行。

[0201] 产业上的可利用性

[0202] 根据本发明,用户可通过动态地配置在单个物理显示装置上提供多个内容的多个屏幕,重新构建存在于系统上的屏幕的形式或连接关系。

[0203] 尽管参照本发明的示例性实施例具体示出并描述了本发明,但是本领域的技术人员应理解:在不脱离由权利要求限定的本发明的精神和范围的情况下,可进行形式和细节上的各种改变。因此,应理解:提供上述示例性实施例仅仅是为了描述的目的,而不是将其解释为对本发明的范围进行任何限制。

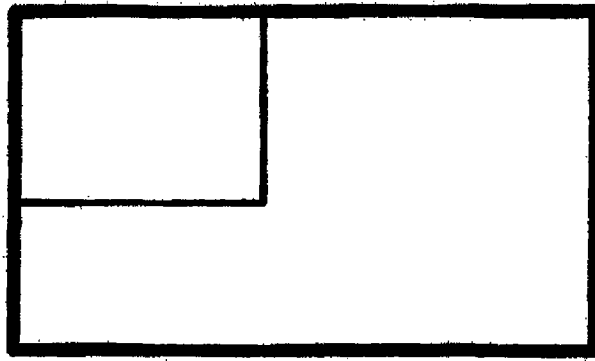


图1A

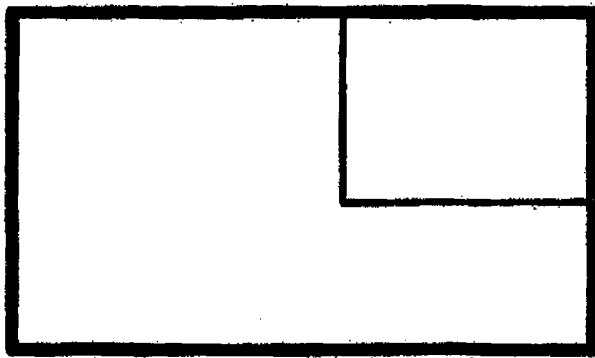


图1B

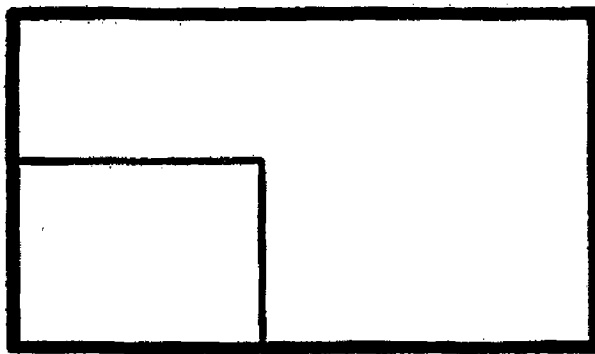


图1C

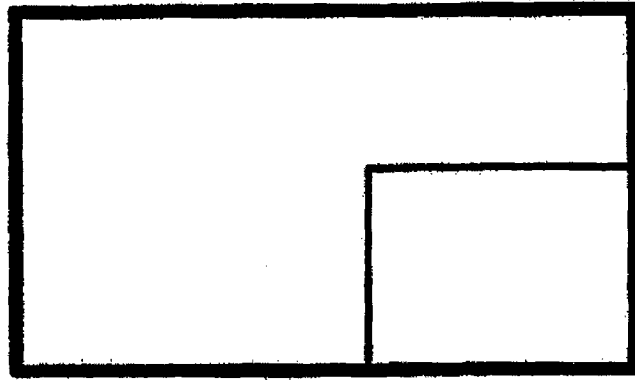


图1D

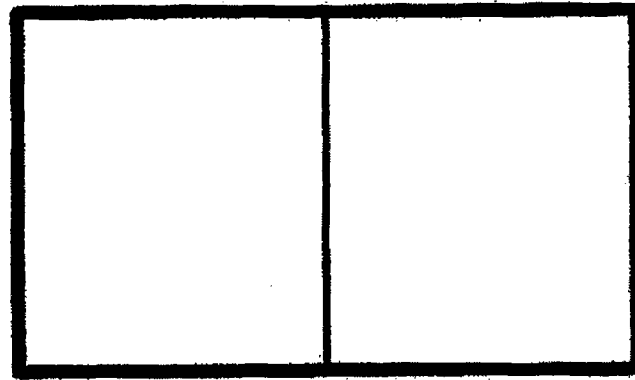


图1E

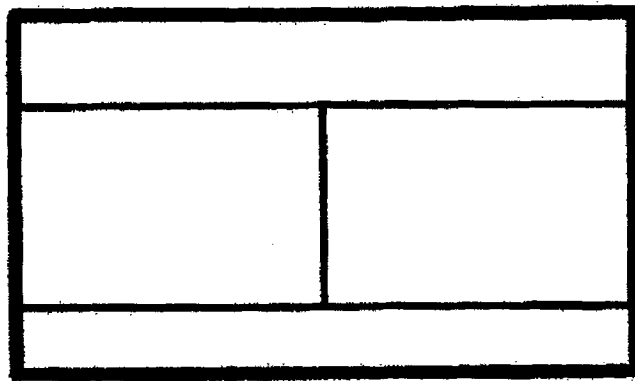


图1F

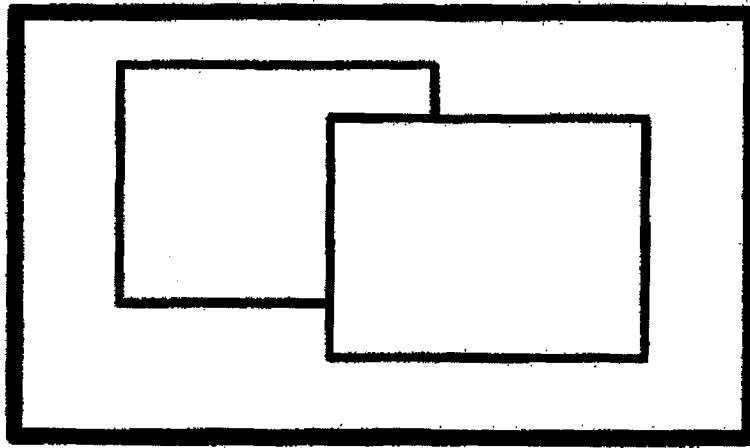


图1G

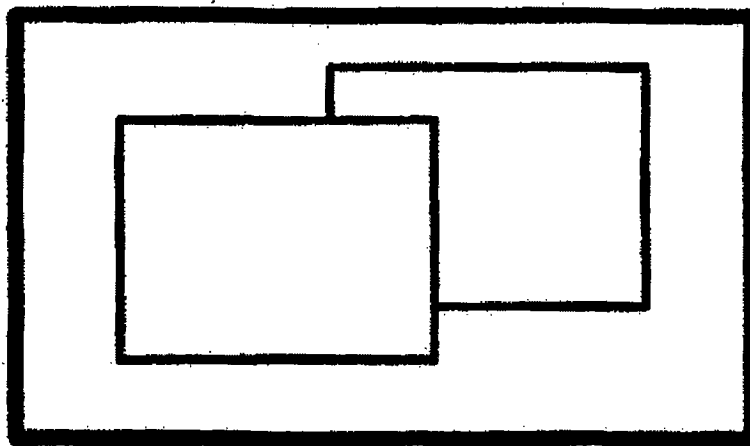


图1H

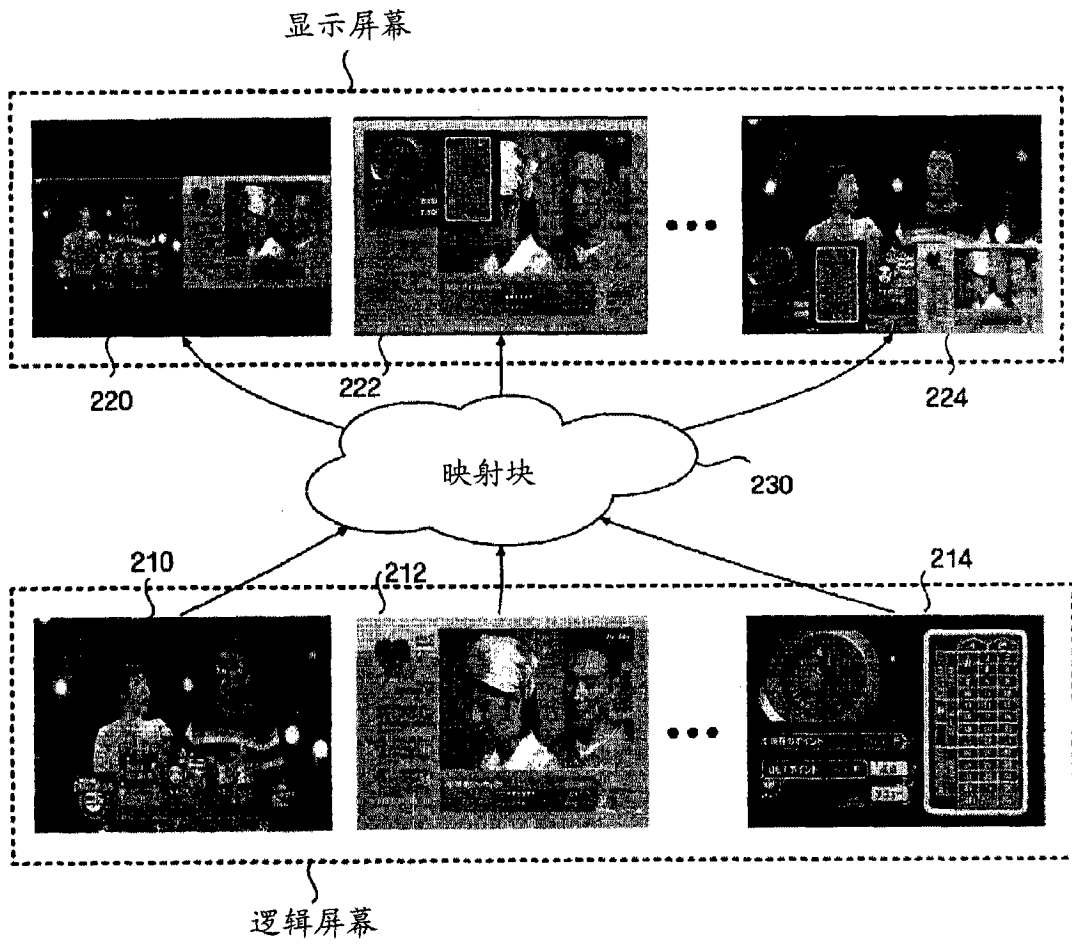


图2

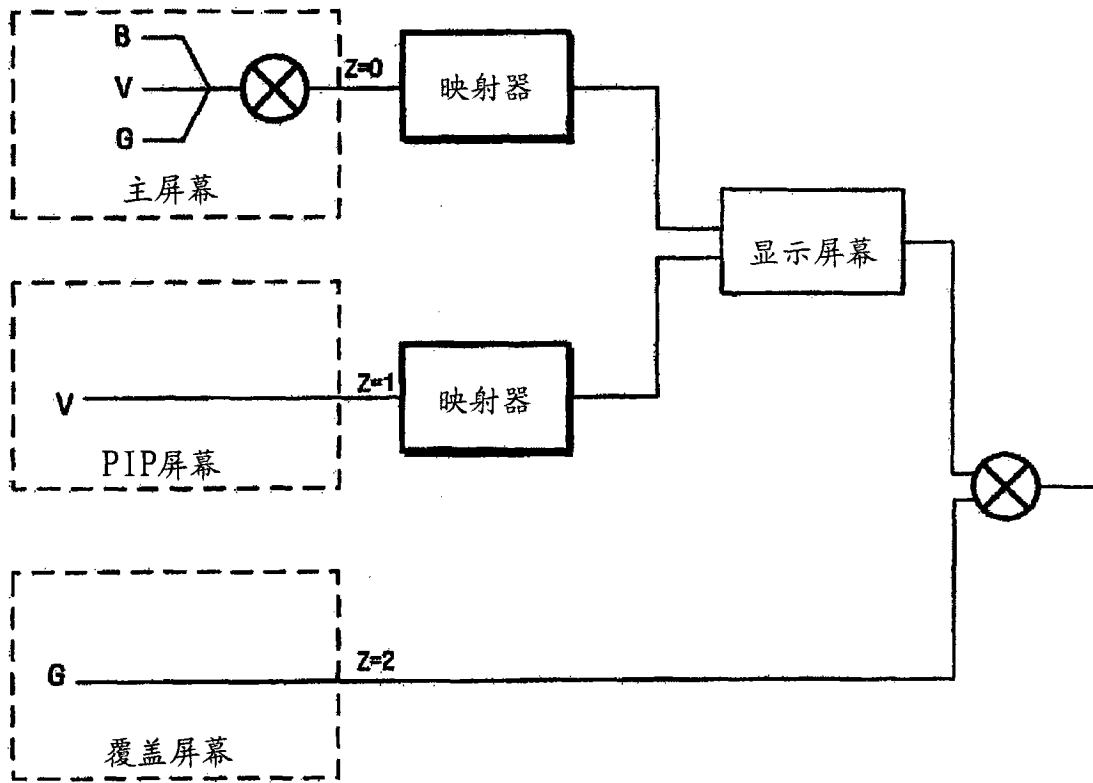


图 3A

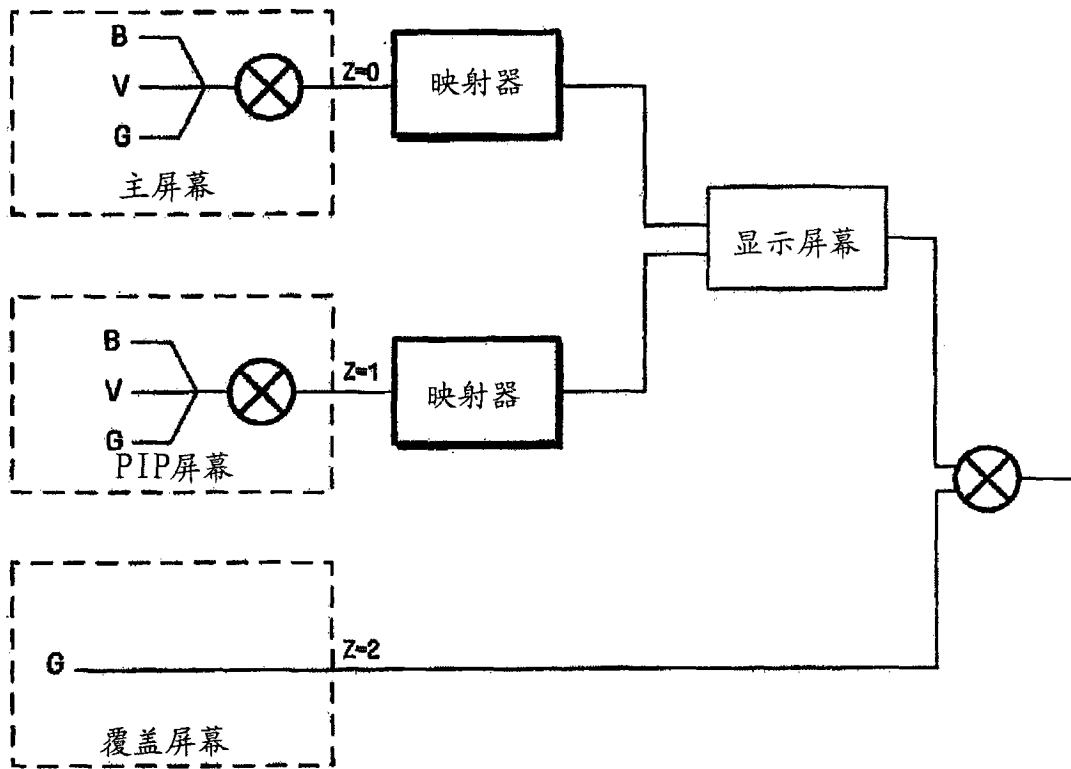


图3B

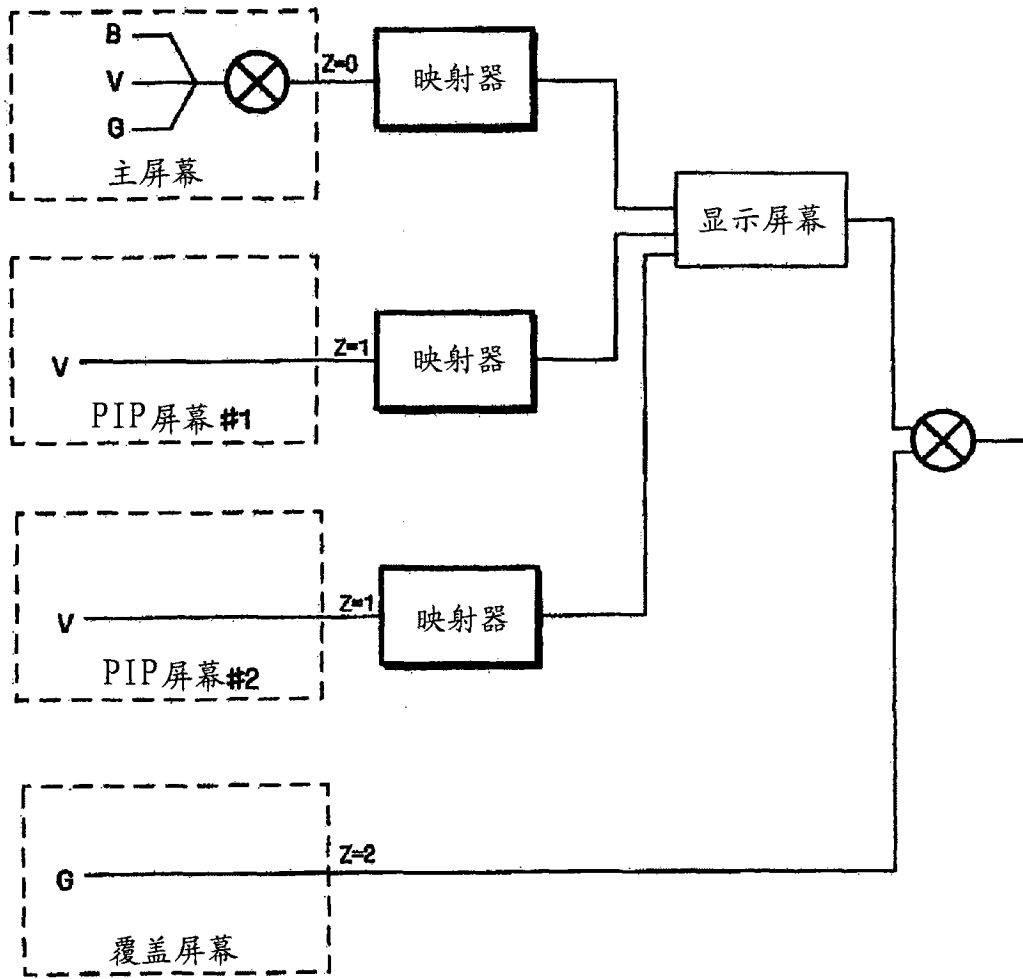


图3C

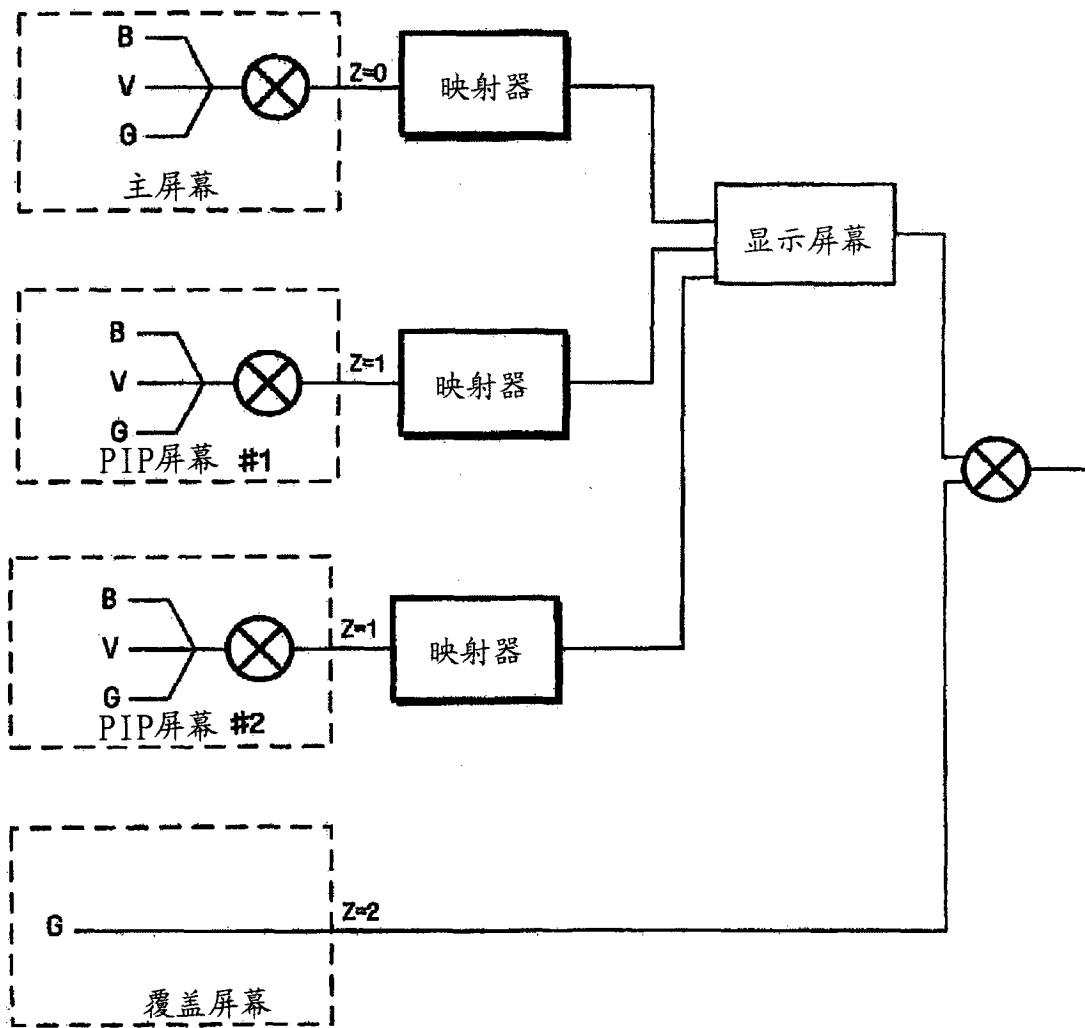


图 3D

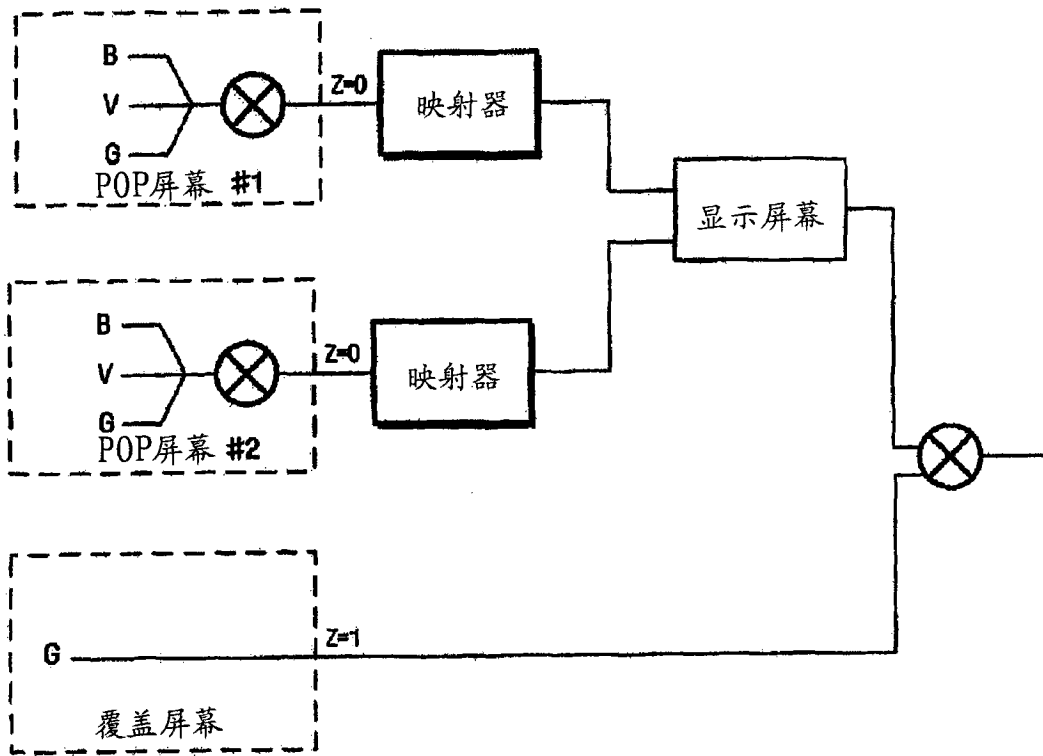


图3E

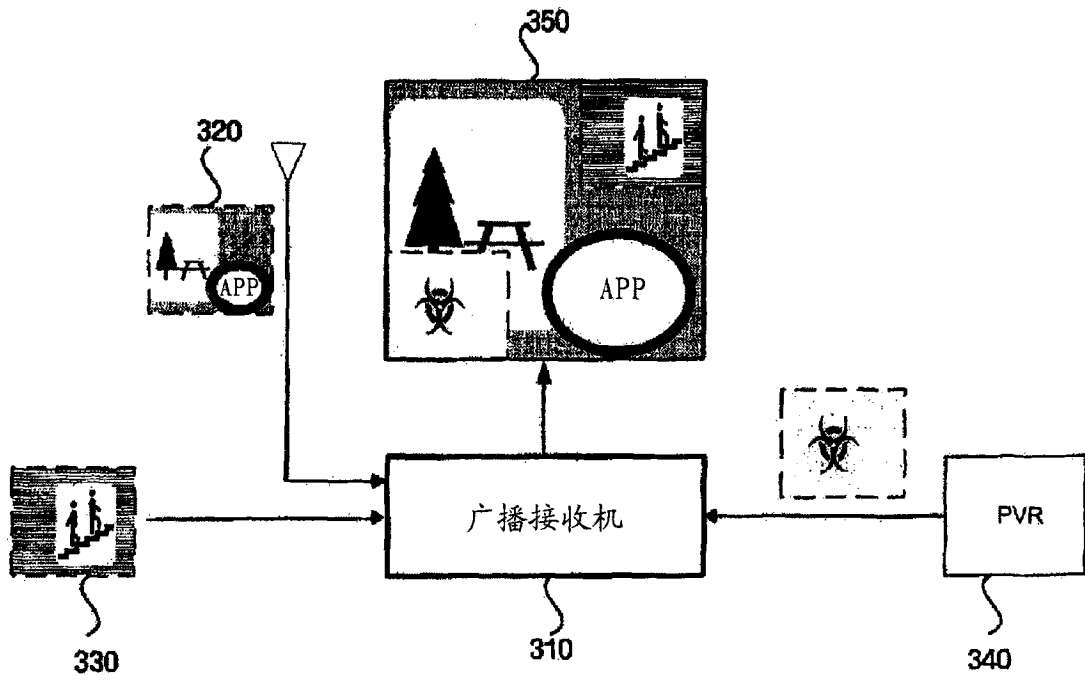


图4

抽象服务

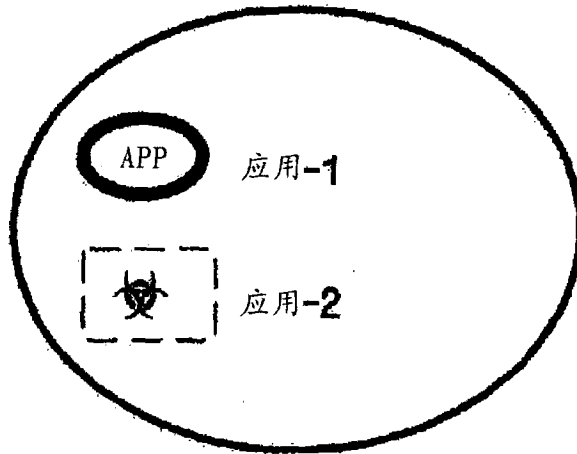


图5A

非抽象服务

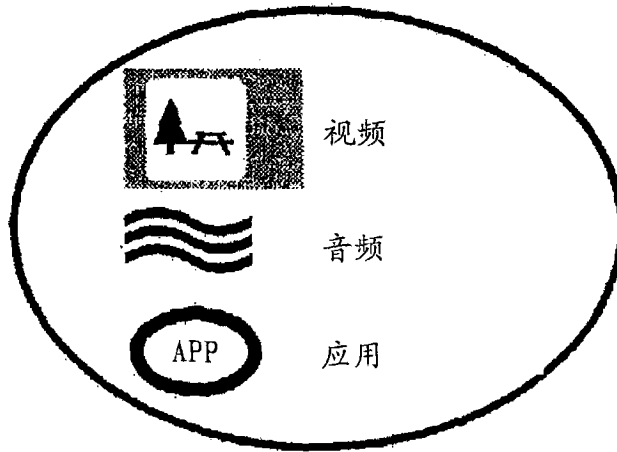


图5B

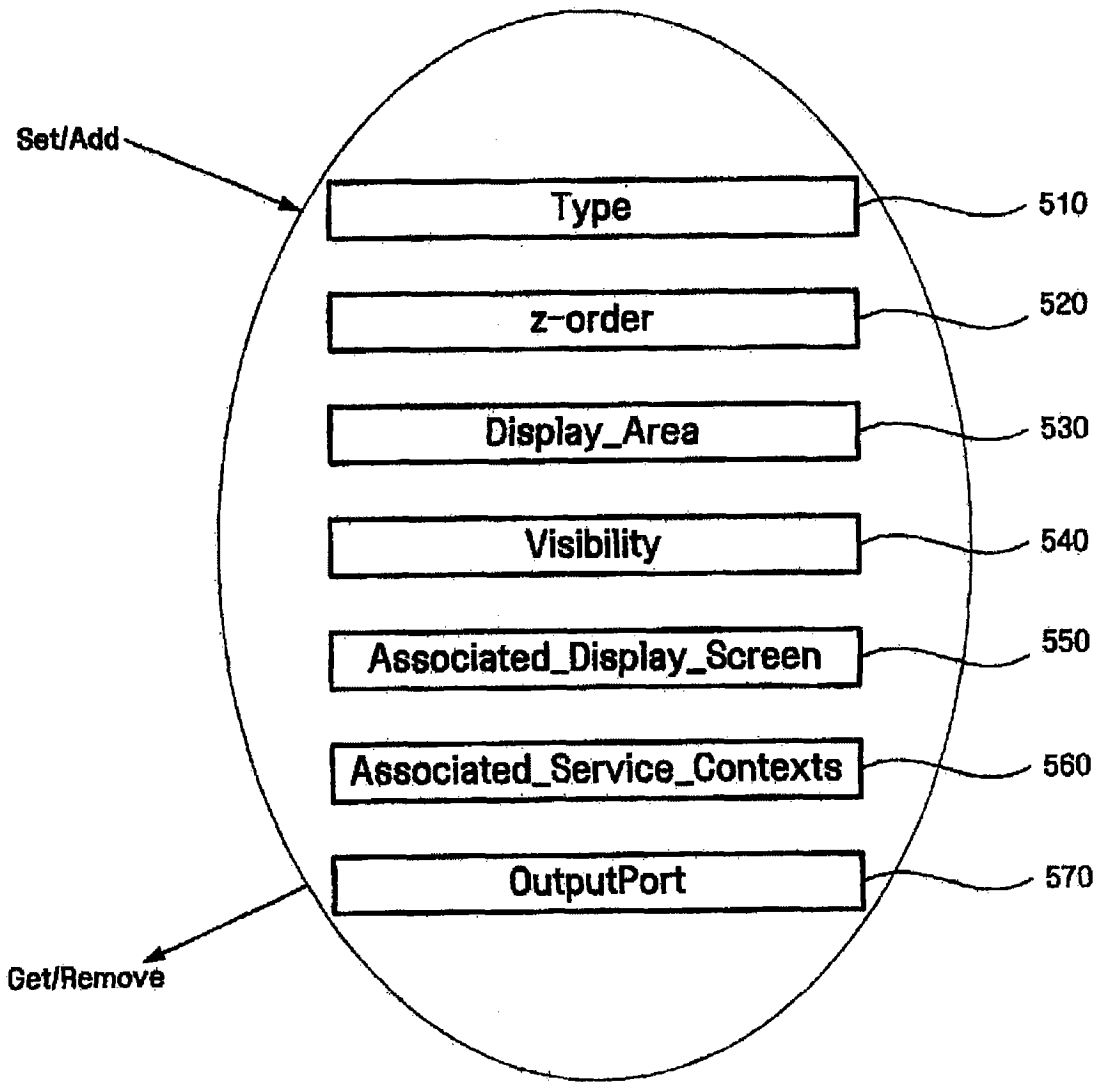


图6

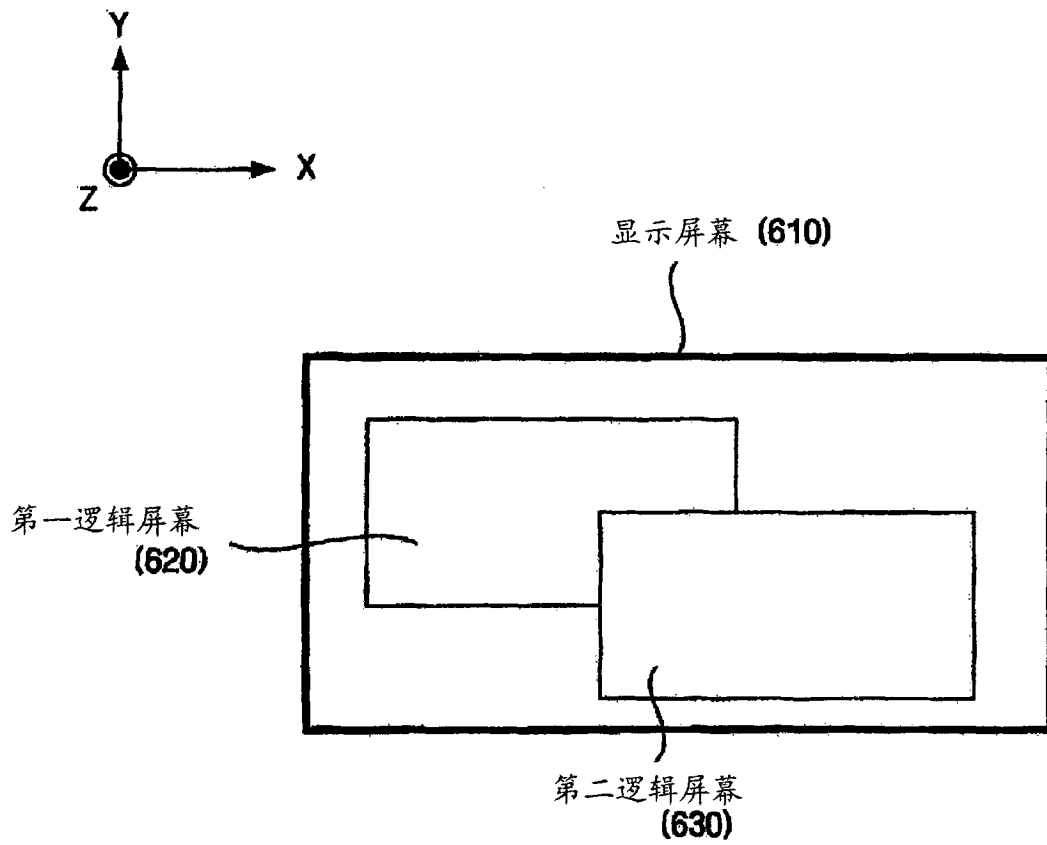


图7

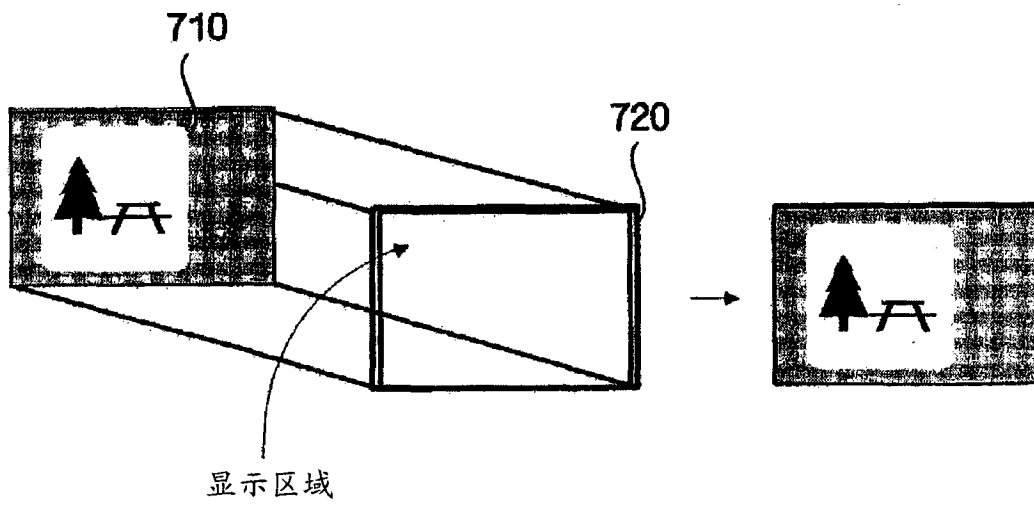


图8A

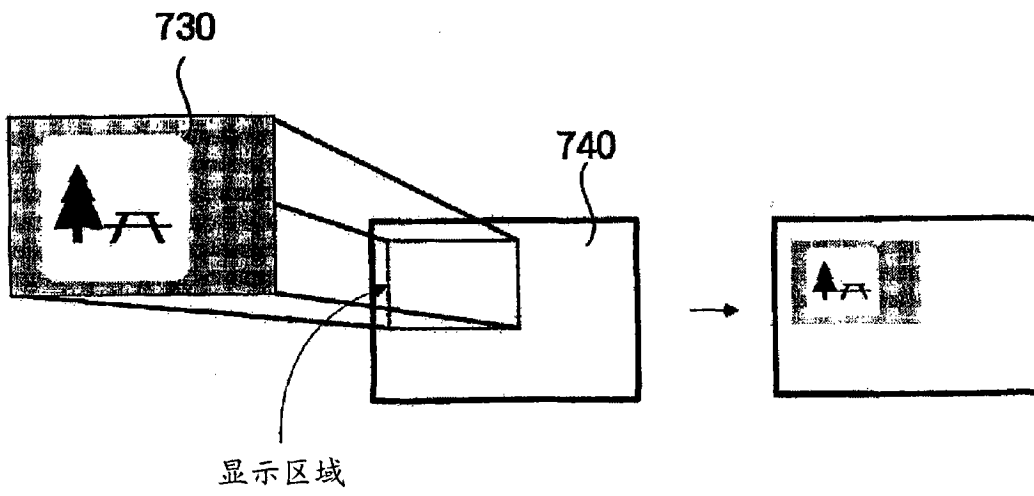


图8B

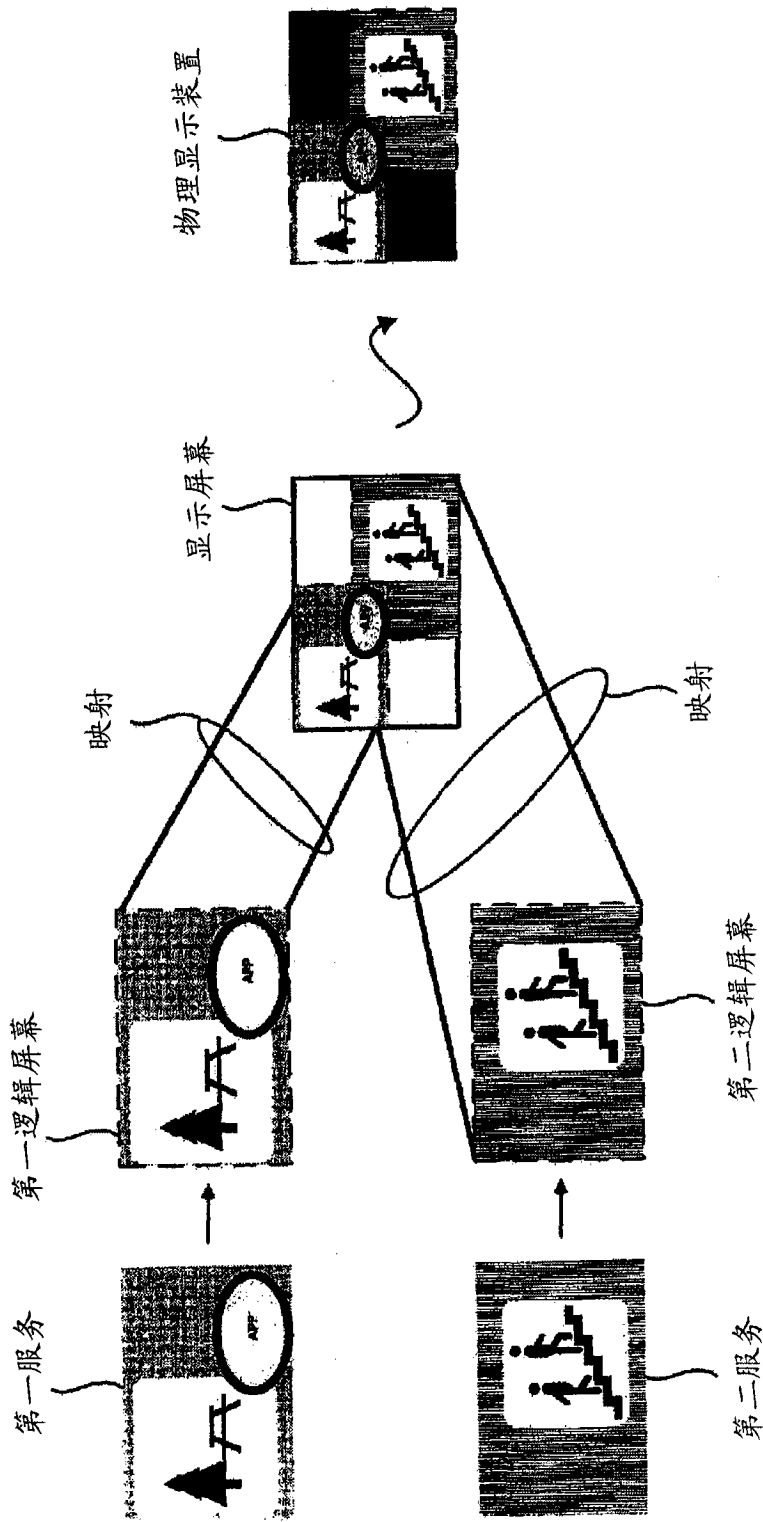


图9

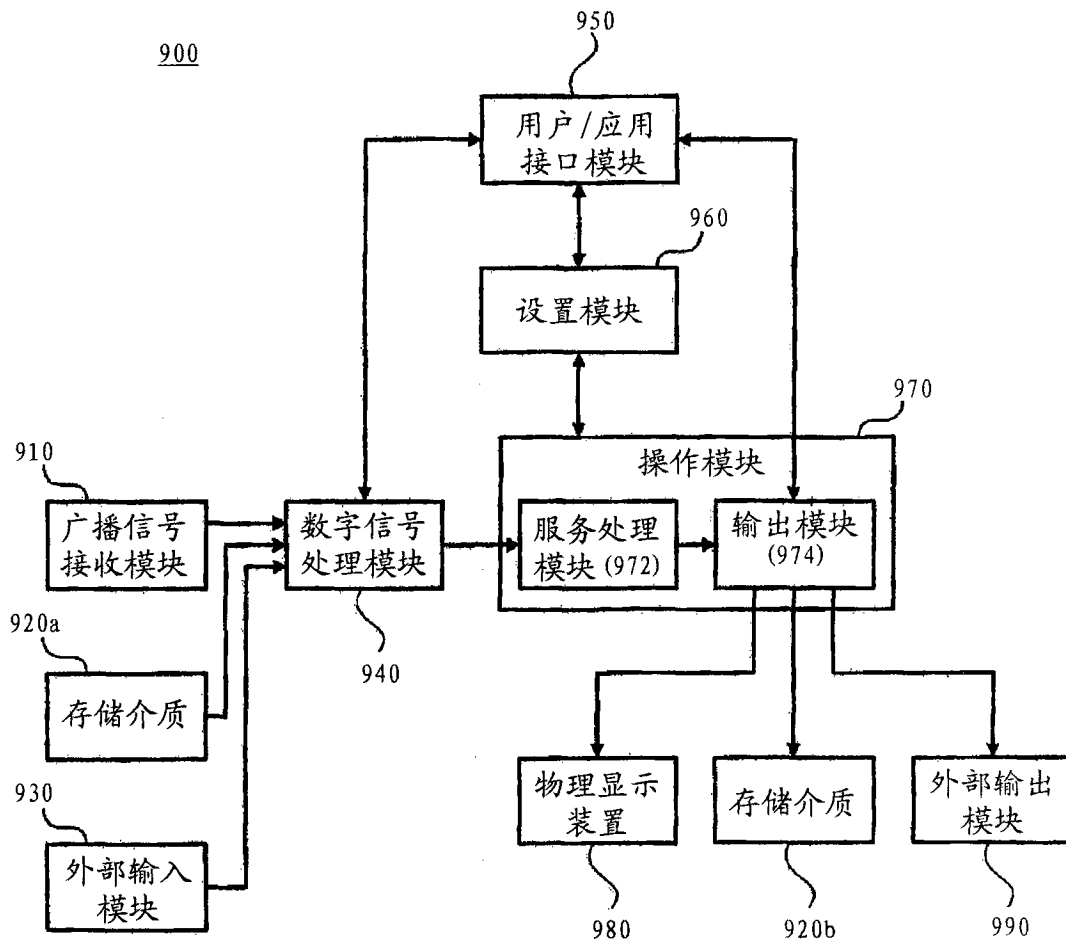


图10

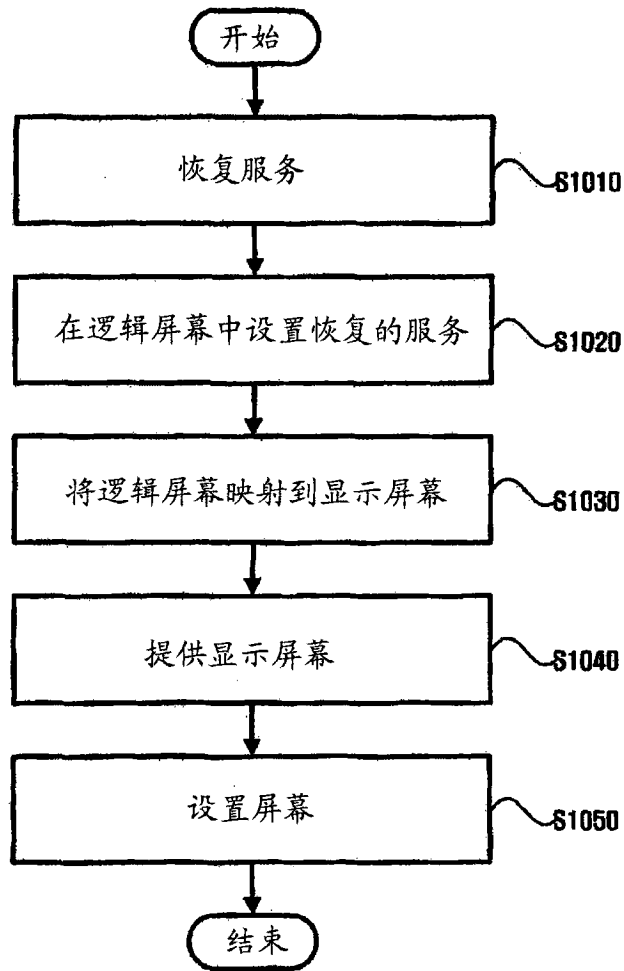


图11

1100

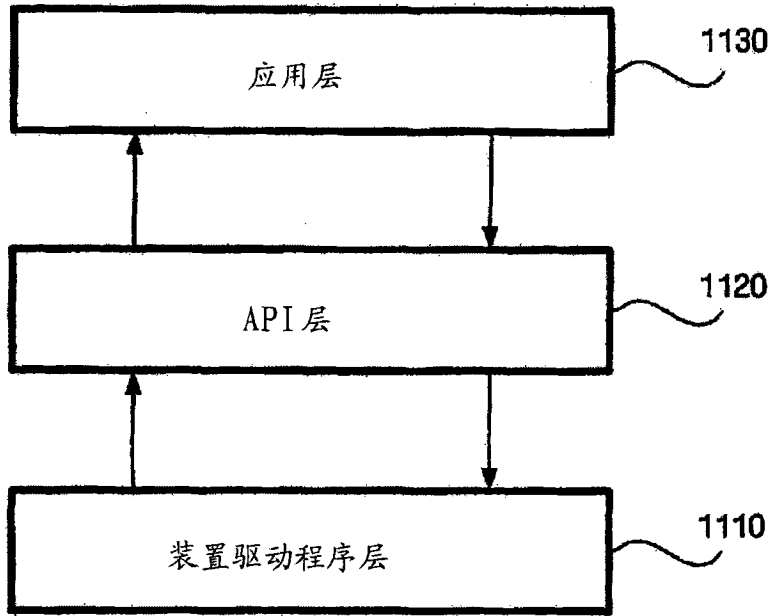


图12

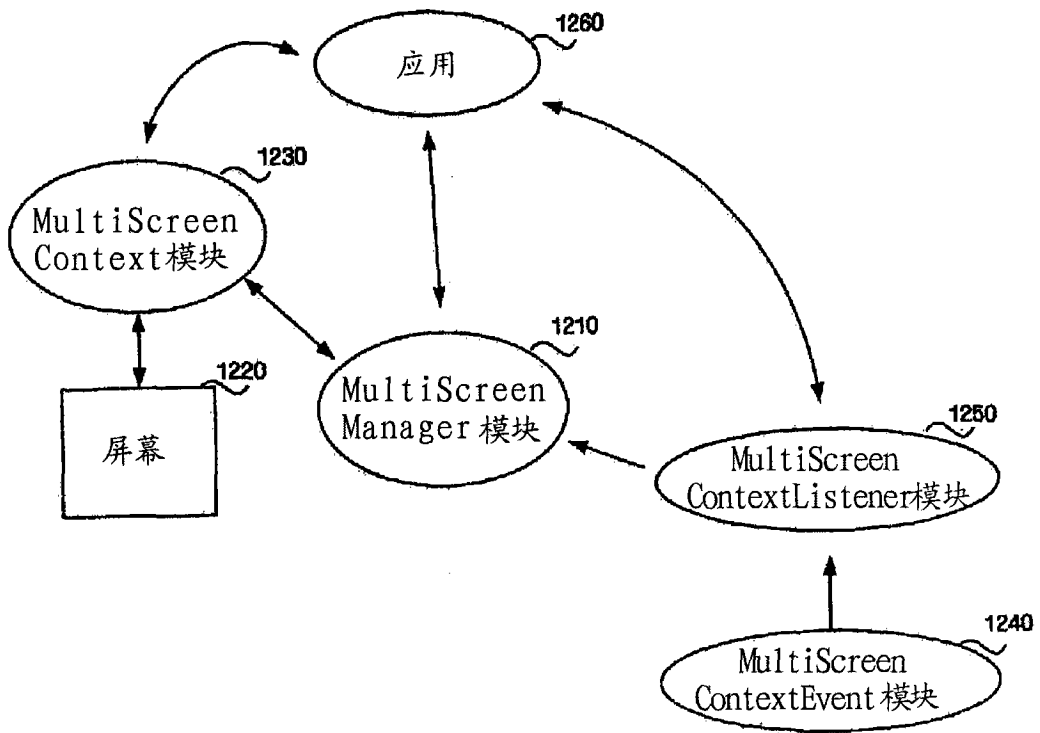


图13

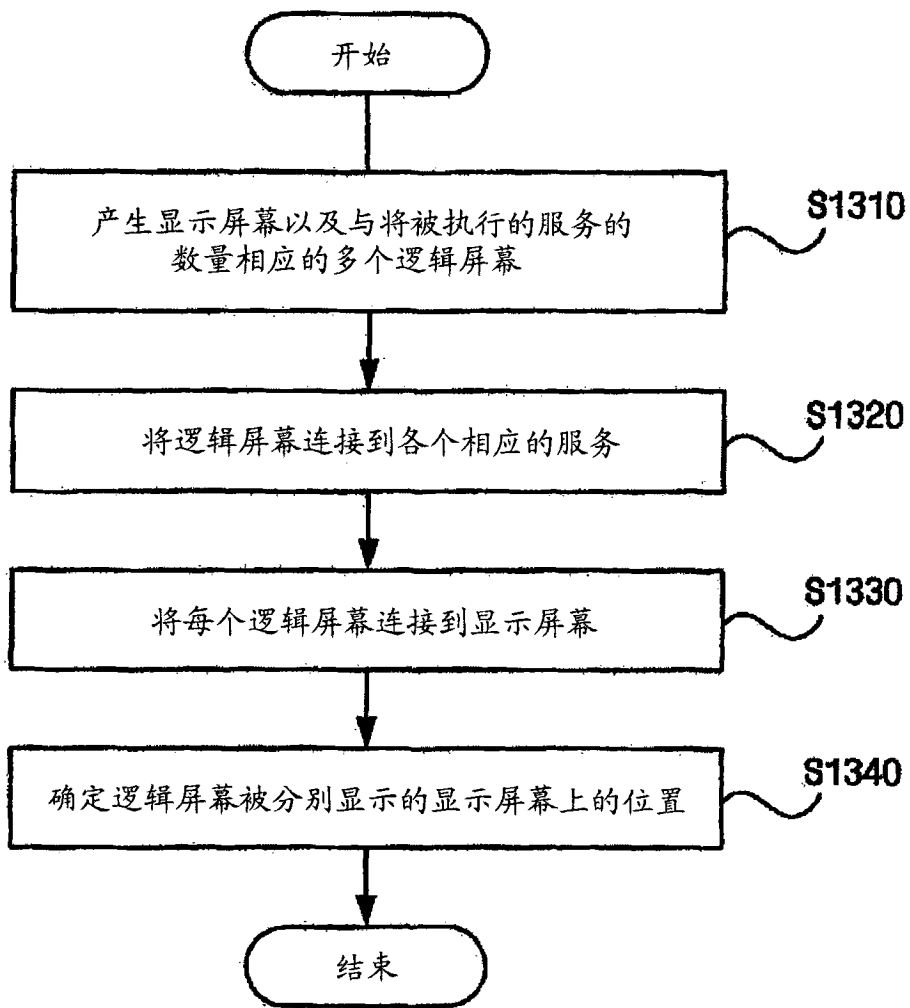


图14

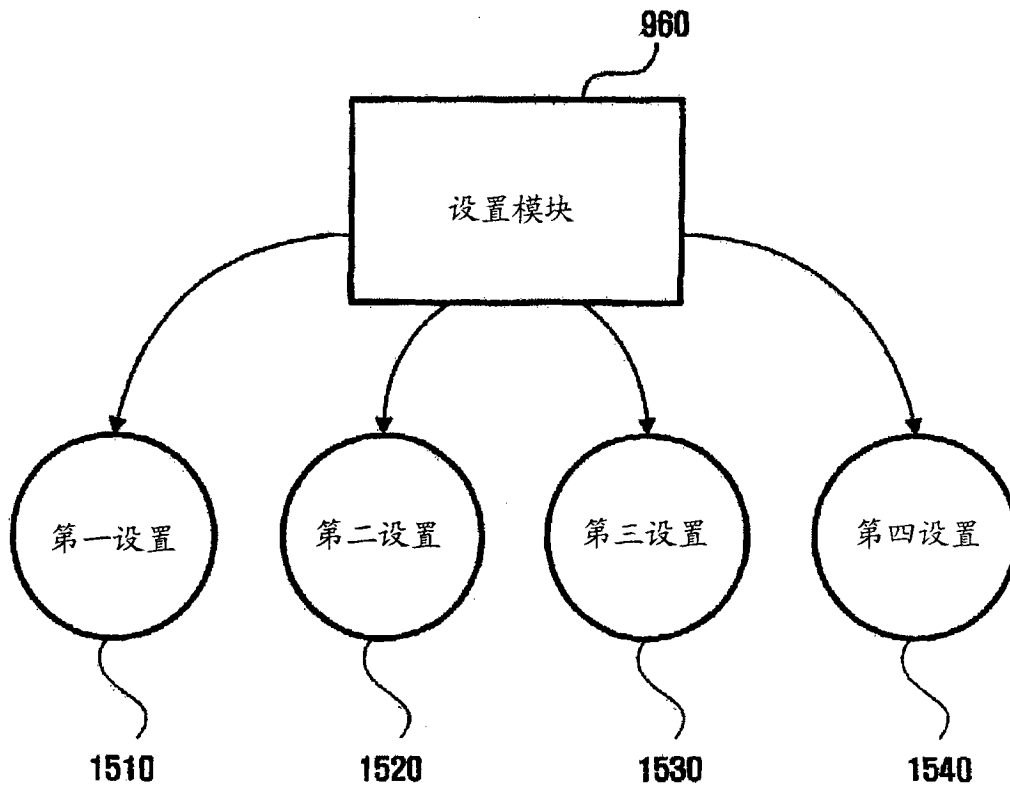


图15

```
public class AContextListener implements MultiScreenContextListener {  
    ...  
  
    public void notify(MultiScreenContextEvent evt) {  
        int id = evt.getId(); // Extract screen in which set is changed  
  
        MultiScreenContext src = (MultiScreenContext) evt.getSource();  
  
        switch (id) {  
            case MultiScreenContextEvent.SCREEN_CHANGED :  
                // Change of basic screen of application  
                break;  
            case MultiScreenContextEvent.SCREEN_CONTEXT_CHANGED :  
                // Change of service contexts connected to screen  
                break;  
            case MultiScreenContextEvent.SCREEN_DISPLAY_SCREEN_CHANGED :  
                // Change of display screen that logical screen is mapped  
                break;  
            case MultiScreenContextEvent.SCREEN_OUTPUT_PORT_CHANGED :  
                // Change of video port  
                break;  
            case MultiScreenContextEvent.SCREEN_VISIBILITY_CHANGED :  
                // Change of screen visibility  
                break;  
            case MultiScreenContextEvent.SCREEN_Z_ORDER_CHANGED :  
                // Change of z-order of screen  
                break;  
            default :  
                break;  
        }  
    }  
    ...  
}  
  
// Generate instance of listener  
AContextListener listener = new AContextListener();  
MultiScreenManager msm = MultiScreenManager.getInstance();  
  
// Register listener  
msm.addScreenContextListener(listener);  
  
...  
  
//Receive change of set of generated screen by listener  
  
...
```

图16A

```
if(screen instanceof MultiScreenConfigurableContext) {
    MultiScreenConfigurableContext configurableScreen =
        (MultiScreenConfigurableContext) screen;

    try {
        HScreenRectangle r = new HScreenRectangle (0.25, 0.25, 0.5, 0.5);
        configurableScreen.setDisplayArea ( r );
    } catch (SecurityException e) {
        // Right is not given
    } catch (IllegalStateException e) {
        // Set is not permitted
    }
} else {
    // Disable to set
}
```

图16B

```
if(screen instanceof MultiScreenConfigurableContext){
    MultiScreenConfigurableContext configurableScreen =
        (MultiScreenConfigurableContext) screen;

    try {
        configurableScreen.setZOrder(3);
    } catch (SecurityException e) {
        // Right is not given
    } catch (IllegalStateException e) {
        // Change of z-order is not permitted
    }
} else {
    // Disable to set
}
```

图16C

```
try {
    MultiScreenManager msm = MultiScreenManager.getInstance();
    MultiScreenConfiguration config = msm.getScreenConfiguration();

    if(config.getScreenConfigurationType() !=
        MultiScreenConfiguration.SCREEN_CONFIGURATION_PIP) {
        MultiScreenConfiguration[] configs = msm.getScreenConfigurations
            (MultiScreenConfiguration.SCREEN_CONFIGURATION_PIP);

        if((configs != null) && (configs[0] != null)) {
            // Select arbitrary first PIP set
            msm.setScreenConfiguration(configs[0]);
        }
    } else {
        // In state of PIP
    }
} catch (SecurityException e) {
    // Right is not given
} catch (RuntimeException e) {
    // Another set state
}
```

图16D

```
// Extract instance of MultiScreenManager
MultiScreenManager msm = MultiScreenManager.getInstance();

// Extract predetermined screen disposing method
MultiScreenConfiguration[] mscs =
Msm.getScreenConfigurations(MultiScreenConfiguration.SCREEN_CONFIGURATION_
POP);

If(mscs != null) {
    MultiScreenConfiguration targetMsc = null;

    for(int i = 0; i < mscs.length; i++)
    {
        // Extract corresponding screen
        HScreen[] screens = mscs[i].getScreens();

        for(int j = 0; j < screens.length; j++)
        {
            // Extract coordinates of each screen, size, or z-order and
            // check whether it is selected by application
            ...
        }

        // (In case of being selected by application)
        {
            targetMsc = mscs[i];
            break;
        }
    }

    If(targetMsc != null)
    {
        // Set screen on the basis of selected disposing method
        msm.setScreenConfiguration(targetMsc);
    }
}
```

图 16E

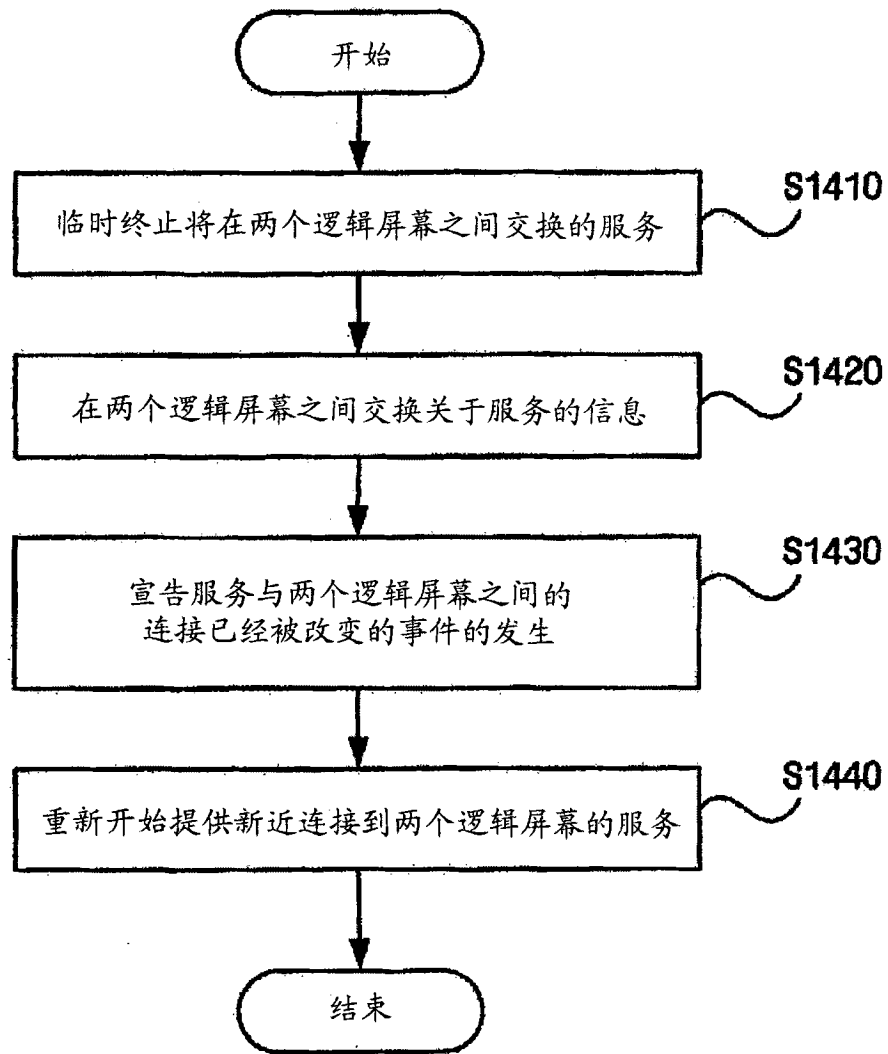


图17