

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 3/048 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200880001811.6

[43] 公开日 2009年11月11日

[11] 公开号 CN 101578577A

[22] 申请日 2008.1.3

[21] 申请号 200880001811.6

[30] 优先权

[32] 2007.1.7 [33] US [31] 11/620,727

[86] 国际申请 PCT/US2008/000060 2008.1.3

[87] 国际公布 WO2008/085848 英 2008.7.17

[85] 进入国家阶段日期 2009.7.7

[71] 申请人 苹果公司

地址 美国加利福尼亚州

[72] 发明人 克里斯多佛·布鲁门伯格

[74] 专利代理机构 北京东方亿思知识产权代理有
限责任公司

代理人 宋鹤南 霆

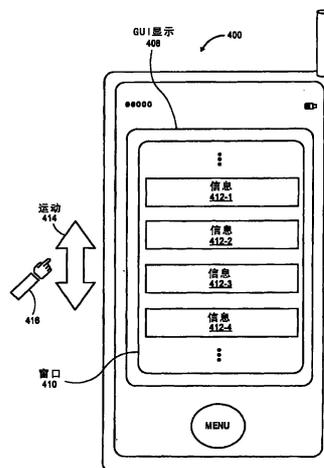
权利要求书 11 页 说明书 33 页 附图 37 页

[54] 发明名称

用于手势操作的应用编程接口

[57] 摘要

本发明的至少某些实施例包括具有与软件应用进行交互的用户接口软件的环境，以给装置的显示器提供手势操作。该环境中通过应用编程接口 (API) 而工作的方法包括传送缩放变换调用。手势操作包括响应于具有两个或更多个输入点的用户输入而执行缩放变换，例如缩小或放大。手势操作还包括响应于具有两个或更多个输入点的用户输入而执行旋转变换以对图像或视图进行旋转。



1. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

传送处理手势事件调用；和

响应于所述处理手势事件调用而传送手势变化调用。

2. 如权利要求 1 所述的方法，其中，响应于与触摸所述显示器的用户输入相关的手势事件而传送所述处理手势事件调用，所述用户输入包括多个输入点。

3. 如权利要求 2 所述的方法，其中，所述手势事件包括和弦事件对象，所述和弦时间对象具有和弦计数、和弦开始事件、和弦变化事件和和弦结束事件。

4. 如权利要求 1 所述的方法，其中，传送所述处理手势事件调用是下列之一：发起、产生、引起或接收所述处理手势事件调用。

5. 如权利要求 1 所述的方法，其中，传送所述手势变化调用是下列之一：发起、产生、引起或接收所述手势变化调用。

6. 如权利要求 1 所述的方法，其中，所述装置是下列之一：数据处理装置、便携装置、便携式数据处理装置、多点触摸装置、多点触摸便携装置、无线装置和蜂窝电话。

7. 一种存储可执行程序指令的计算机可读介质，所述可执行程序指令在执行时使得数据处理系统执行一种方法，该方法包括：

传送处理手势事件调用；和

响应于所述处理手势事件调用而传送手势变化调用。

8. 如权利要求 7 所述的计算机可读介质，其中，响应于与触摸所述显示器的用户输入相关的手势事件而传送所述处理手势事件调用，所述用户输入包括多个输入点。

9. 如权利要求 7 所述的计算机可读介质，其中，所述手势对象包括和弦事件对象，所述和弦事件对象具有和弦计数、和弦开始事件、和弦变

化事件和和弦结束事件。

10. 如权利要求 7 所述的计算机可读介质，其中，传送所述处理手势事件调用是下列之一：发起、产生、引起或接收所述处理手势事件调用。

11. 如权利要求 7 所述的计算机可读介质，其中，传送所述手势变化调用是下列之一：发起、产生、引起或接收所述手势变化调用。

12. 如权利要求 7 所述的计算机可读介质，其中，所述装置是下列之一：数据处理装置、便携装置、便携式数据处理装置、多点触摸装置、多点触摸便携装置、无线装置和蜂窝电话。

13. 一种设备，该设备用于下述环境中：该环境带有与软件应用交互的用户接口软件，该设备包括：

传送处理手势事件调用的装置；和

响应于所述处理手势事件调用而传送手势变化调用的装置。

14. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

接收处理手势事件调用；和

响应于所述处理手势事件调用而接收手势变化调用。

15. 如权利要求 14 所述的方法，其中，响应于与触摸所述显示器的用户输入相关的手势事件而接收所述处理手势事件调用，所述用户输入包括多个输入点。

16. 如权利要求 14 所述的方法，其中，所述手势事件包括和弦事件对象，所述和弦事件对象具有和弦计数、和弦开始事件、和弦变化事件和和弦结束事件。

17. 一种存储可执行程序指令的计算机可读介质，所述可执行程序指令在执行时使得数据处理系统执行一种方法，该方法包括：

接收处理手势事件调用；和

响应于所述处理手势事件调用而接收手势变化调用。

18. 如权利要求 17 所述的方法，其中，响应于与触摸所述显示器的用户输入相关的手势事件而接收所述处理手势事件调用，所述用户输入包

括多个输入点。

19. 如权利要求 18 所述的方法，其中，所述手势事件包括和弦事件对象，所述和弦事件对象具有和弦计数、和弦开始事件、和弦变化事件和和弦结束事件。

20. 一种设备，该设备用于下述环境中：该环境带有与软件应用交互的用户接口软件，该设备包括：

接收处理手势事件调用的装置；和

响应于所述处理手势事件调用而接收手势变化调用的装置。

21. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

传送缩放变换调用，以确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

22. 如权利要求 21 所述的方法，还包括：

传送开始缩放手势调用；

传送缩放手势进展调用；和

传送缩放手势结束调用。

23. 如权利要求 21 所述的方法，其中，所述缩放变换包括最小和最大缩放因子。

24. 如权利要求 21 所述的方法，其中，所述缩放变换基于被禁用的标记而从最小缩放因子变化到最大缩放因子。

25. 如权利要求 21 所述的方法，其中，所述缩放变换基于被允许的标记而在快速返回到最大缩放因子之前从最小缩放因子变化到略超过最大缩放因子。

26. 如权利要求 21 所述的方法，其中，传送所述缩放变换调用是下列之一：发起、产生、引起或接收所述缩放变换调用。

27. 如权利要求 21 所述的方法，其中，所述装置是下列之一：数据处理装置、便携装置、便携式数据处理装置、多点触摸装置、多点触摸便携装置、无线装置和蜂窝电话。

28. 一种存储可执行程序指令的计算机可读介质，所述可执行程序指令在执行时使得数据处理系统执行一种方法，该方法包括：

传送缩放变换调用，以确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

29. 如权利要求 28 所述的计算机可读介质，还包括：

传送开始缩放手势调用；

传送缩放手势进展调用；和

传送缩放手势结束调用。

30. 如权利要求 28 所述的计算机可读介质，其中，所述缩放变换包括最小和最大因子。

31. 如权利要求 28 所述的计算机可读介质，其中，所述缩放变换基于被禁用的标记而从最小缩放因子变化到最大缩放因子。

32. 如权利要求 28 所述的计算机可读介质，其中，所述缩放变换基于被允许的标记而在快速返回到最大缩放因子之前从最小缩放因子变化到略超过最大缩放因子。

33. 如权利要求 28 所述的计算机可读介质，其中，传送所述缩放变换调用是下列之一：发起、产生、引起或接收所述缩放变换调用。

34. 如权利要求 28 所述的计算机可读介质，其中，所述数据处理系统是下列之一：数据处理装置、便携装置、便携式数据处理装置、多点触摸装置、多点触摸便携装置、无线装置和蜂窝电话。

35. 一种设备，该设备用于下述环境中：该环境带有与软件应用交互的用户接口软件，该设备包括：

传送缩放变换调用的装置，所述缩放变换调用确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

36. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

接收缩放变换调用，以确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

37. 如权利要求 36 所述的方法，还包括：

接收开始缩放手势调用；

接收缩放手势进展调用；和

接收缩放手势结束调用。

38. 如权利要求 36 所述的方法，其中，所述缩放变换包括最小和最大因子。

39. 如权利要求 36 所述的方法，其中，所述缩放变换基于被禁用的标记而从最小缩放因子变化到最大缩放因子。

40. 如权利要求 36 所述的方法，其中，所述缩放变换基于被允许的标记而在快速返回到最大缩放因子之前从最小缩放因子变化到略超过最大缩放因子。

41. 一种存储可执行程序指令的计算机可读介质，所述可执行程序指令在执行时使得数据处理系统执行一种方法，该方法包括：

接收缩放变换调用，以确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

42. 如权利要求 36 所述的计算机可读介质，还包括：

接收开始缩放手势调用；

接收缩放手势进展调用；和

接收缩放手势结束调用。

43. 如权利要求 36 所述的计算机可读介质，其中，所述缩放变换包括最小和最大因子。

44. 如权利要求 36 所述的计算机可读介质，其中，所述缩放变换基于被禁用的标记而从最小缩放因子变化到最大缩放因子。

45. 如权利要求 36 所述的计算机可读介质，其中，所述缩放变换基于被允许的标记而在快速返回到最大缩放因子之前从最小缩放因子变化到略超过最大缩放因子。

46. 一种设备，该设备用于下述环境中：该环境带有与软件应用交互的用户接口软件，该设备包括：

接收缩放变换调用的装置，所述缩放变换调用确定用于与用户输入相

关的视图的缩放变换，所述用户输入具有多个输入点。

47. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

设定手势缩放变换标记，以确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

48. 如权利要求 47 所述的方法，其中，所述缩放变换标记或者从当前缩放因子缩放到最小缩放因子，或者从当前缩放因子缩放到最大缩放因子。

49. 一种存储可执行程序指令的计算机可读介质，所述可执行程序指令在执行时使得数据处理系统执行一种方法，该方法包括：

设定手势缩放变换标记，以确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

50. 如权利要求 49 所述的计算机可读介质，其中，所述缩放变换标记或者从当前缩放因子缩放到最小缩放因子，或者从当前缩放因子缩放到最大缩放因子。

51. 一种设备，该设备用于下述环境中：该环境带有与软件应用交互的用户接口软件，该设备包括：

设定手势缩放变换标记的装置，所述手势缩放变换标记确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

52. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

传送旋转变换调用，以确定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

53. 如权利要求 52 所述的方法，还包括：

传送开始旋转手势调用；

传送旋转手势进展调用；和

传送旋转手势结束调用。

54. 如权利要求 52 所述的方法，其中，所述旋转变换包括用于相关的最小和最大旋转视图的最小和最大旋转角度。

55. 如权利要求 52 所述的方法，其中，传送所述旋转变换调用是下列之一：发起、产生、引起或接收所述旋转变换调用。

56. 如权利要求 52 所述的方法，其中，所述装置是下列之一：数据处理装置、便携装置、便携式数据处理装置、多点触摸装置、多点触摸便携装置、无线装置和蜂窝电话。

57. 一种存储可执行程序指令的计算机可读介质，所述可执行程序指令在执行时使得数据处理系统执行一种方法，该方法包括：

传送旋转变换调用，以确定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

58. 如权利要求 57 所述的计算机可读介质，还包括：

传送开始旋转手势调用；

传送旋转手势进展调用；和

传送旋转手势结束调用。

59. 如权利要求 57 所述的计算机可读介质，其中，所述旋转变换包括用于相关的最小和最大旋转视图的最小和最大旋转角度。

60. 如权利要求 57 所述的计算机可读介质，其中，传送所述旋转变换调用是下列之一：发起、产生、引起或接收所述旋转变换调用。

61. 如权利要求 57 所述的计算机可读介质，其中，所述数据处理系统是下列之一：数据处理装置、便携装置、便携式数据处理装置、多点触摸装置、多点触摸便携装置、无线装置和蜂窝电话。

62. 一种设备，该设备用于下述环境中：该环境带有与软件应用交互的用户接口软件，该设备包括：

传送旋转变换调用的装置，所述旋转变换调用确定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

63. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

接收旋转变换调用，以确定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

64. 如权利要求 63 所述的方法，还包括：

接收开始旋转手势调用；

接收旋转手势进展调用；和

接收旋转手势结束调用。

65. 如权利要求 36 所述的方法，其中，所述旋转变换包括用于相关的最小和最大旋转视图的最小和最大旋转角度。

66. 一种存储可执行程序指令的计算机可读介质，所述可执行程序指令在执行时使得数据处理系统执行一种方法，该方法包括：

接收旋转变换调用，以确定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

67. 如权利要求 66 所述的计算机可读介质，还包括：

接收开始旋转手势调用；

接收旋转手势进展调用；和

接收旋转手势结束调用。

68. 如权利要求 66 所述的计算机可读介质，其中，所述旋转变换包括用于相关的最小和最大旋转视图的最小和最大旋转角度。

69. 一种设备，该设备用于下述环境中：该环境带有与软件应用交互的用户接口软件，该设备包括：

接收旋转变换调用的装置，所述旋转变换调用确定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

70. 一种通过应用编程接口（API）操作的方法，该方法用于下述装置的环境中：该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件，该方法包括：

传送委托调用，以向委托通知手势事件。

71. 如权利要求 70 所述的方法，其中，所述委托调用向所述委托通知与视图上的缩放变换相关的手势事件，以便在所述软件应用的控制下操作的所述委托使其它视图响应于由所述手势事件修改的视图而变化。

72. 如权利要求 70 所述的方法, 其中, 传送所述委托调用是下列之一: 发起、产生、引起或接收所述委托调用。

73. 一种存储可执行程序指令的计算机可读介质, 所述可执行程序指令在执行时使得数据处理系统执行一种方法, 该方法包括:

传送委托调用, 以向委托通知手势事件。

74. 如权利要求 73 所述的计算机可读介质, 其中, 所述委托调用向所述委托通知与视图上的缩放变换相关的手势事件, 以便在所述软件应用的控制下操作的所述委托使其它视图响应于由所述手势事件修改的视图而变化。

75. 如权利要求 73 所述的方法, 其中, 传送所述委托调用是下列之一: 发起、产生、引起或接收所述委托调用。

76. 一种设备, 该设备用于下述环境中: 该环境带有与软件应用交互的用户接口软件, 该设备包括:

接收委托调用的装置, 所述委托调用用于向委托通知手势事件。

77. 一种通过应用编程接口 (API) 操作的方法, 该方法用于下述装置的环境中: 该环境带有具有多视图的显示器以及与软件应用交互的用户接口软件, 该方法包括:

接收委托调用, 以向委托通知手势事件。

78. 如权利要求 77 所述的方法, 其中, 所述委托调用向委托通知与视图上的缩放变换相关的手势事件, 以便于在所述软件应用的控制下操作的所述委托使其它视图响应于由所述手势事件修改的视图而变化。

79. 一种存储可执行程序指令的计算机可读介质, 所述可执行程序指令在执行时使得数据处理系统执行一种方法, 该方法包括:

接收委托调用, 以向委托通知手势事件。

80. 如权利要求 79 所述的计算机可读介质, 其中, 所述委托调用向委托通知与视图上的缩放变换相关的手势事件, 以便于在所述软件应用的控制下操作的所述委托使其它视图响应于由所述手势事件修改的视图而变化。

81. 一种设备, 该设备用于下述环境中: 该环境带有与软件应用交互

的用户接口软件，该设备包括：

接收委托调用的装置，所述委托调用用于向委托通知手势事件。

82. 一种装置，包括：

输入面板，其配置为以两个或多个输入点的形式接收用户输入；

显示装置，其与所述输入面板集成为一体；

中央处理单元，其耦联到所述输入面板；

存储器，其耦联到所述处理器；

一个或多个程序，其中，所述一个或多个程序存储在存储器中并配置为由所述一个或多个处理器执行，所述程序包括：

用于对所述输入面板上或所述输入面板附近用户输入的运动进行检测的指令；

用于响应于检测所述用户输入而处理与视图相关的手势事件的指令。

83. 如权利要求 82 所述的装置，还包括用于执行与所述视图相关的缩放变换的指令。

84. 如权利要求 82 所述的装置，其中，所述处理器配置为基于所述输入面板的每个触点的正电荷和负电荷来确定哪个视图是所述用户输入想要选择的。

85. 如权利要求 82 所述的装置，其中，所述装置是下列之一：数据处理装置、便携装置、便携式数据处理装置、多点触摸装置、多点触摸便携装置、无线装置和蜂窝电话。

86. 一种包含可执行指令的计算机可读介质，包括：

程序代码，其配置为在执行时通过应用编程接口传送缩放变换调用，所述缩放变换调用确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

87. 如权利要求 86 所述的计算机可读介质，其中，传送所述缩放变换调用是下列之一：发起、产生、引起或接收所述缩放变换调用。

88. 一种包含可执行指令的计算机可读介质，包括：

程序代码，其配置为在执行时通过应用编程接口传送旋转变换调用，所述旋转变换调用确定用于与用户输入相关的视图的旋转变换，所述用户

输入具有多个输入点。

89. 如权利要求 88 所述的计算机可读介质，其中，传送所述旋转变换调用是下列之一：发起、产生、引起或接收所述旋转变换调用。

90. 一种用于执行滚动操作的平台，包括：

处理单元，其耦联到输入面板；

存储器，其耦联到所述处理单元；和

一个或多个程序，其中，所述一个或多个程序存储在所述存储器中，并配置为由所述处理单元执行，所述程序包括：

用于通过应用编程接口传送缩放变换调用的指令，所述缩放变换调用确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

91. 如权利要求 90 所述的平台，其中，所述一个或多个程序还包括用于通过应用编程接口传送旋转变换调用的指令，所述旋转变换调用确定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

92. 如权利要求 90 所述的平台，其中，传送所述缩放变换调用是下列之一：发起、产生、引起或接收所述缩放变换调用。

93. 如权利要求 91 所述的平台，其中，传送所述旋转变换调用是下列之一：发起、产生、引起或接收所述旋转变换调用。

94. 一种用于执行滚动操作的构架，包括：

目录，其用于存储一个或多个程序及相关的资源，其中，所述程序包括下述指令：所述指令在执行时提供应用编程接口以指定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。

95. 如权利要求 94 所述的构架，其中，所述一个或多个程序还包括下述指令：所述指令在执行时提供应用编程接口以指定用于与用户输入相关的视图的旋转变换，所述用户输入具有多个输入点。

用于手势操作的应用编程接口

技术领域

[1]本发明涉及提供手势（gesture）操作的应用编程接口。

计算机程序列表

[2]本专利文献公开内容的一部分包含受到版权保护的材料。版权拥有者不反对任何人以本专利文献或专利公开在专利商标局文档或记录中出现的形式对其进行复制，但是在其他任何方面保留所有的版权。

[3]申请人在此提交所附的附录 A 中所包括的计算机程序列表。

背景技术

[4]API 是计算机系统或程序库提供的源代码接口，用于支持来自软件应用的服务请求。API 在含义方面解释为：能够在建立应用时被解释或编译的编程语言，而不是对数据如何被布置在存储器中进行的显式低层描述。提供了由 API 所描述的功能的软件被称为 API 的实现形式。

[5]各种装置（例如电子装置、计算系统、便携式装置以及手持式装置）具有软件应用。API 在软件应用与用户接口软件之间作为接口，以给装置的用户提供特定的特性和操作。用户可能希望对于装置的显示器有特定的操作，例如滚动（scrolling）、选择、手势和动画操作。

[6]滚动是使内容（例如文本、图画或图像）的方向性（例如水平或垂直）展示形式在屏幕或显示窗口上进行滑动的动作。在通常的图形用户接口中，滚动是在滚动条的帮助下或者利用键盘快捷方式（通常是箭头键）来进行的。手势是具有两个或更多个输入点的用户输入类型。动画操作包括在给定的时间段内改变内容。

[7]各种装置可能具有有限的显示器尺寸、用户接口、软件、API 接口和/或处理能力，这限制了装置的易用性。装置的用户接口实现 API 以提供所

请求的功能和特性。这些用户接口可能难以解释各种用户输入并提供与这些用户输入相关的所需功能。

发明内容

[8]本发明的至少某些实施例包括下述环境中的一个或多个应用编程接口：该环境具有与软件应用进行交互的用户接口软件。各种函数调用或消息通过这些应用编程接口而在用户接口软件与软件应用之间传送。示例性的应用编程接口传送函数调用以实现装置的滚动、手势和动画操作。

[9]本发明的至少某些实施例包括具有与软件应用进行交互的用户接口软件的环境。该环境中通过应用编程接口（API）而工作的方法包括传送设定弹回调用。该方法还包括设定最大和最小弹回值中至少一者。设定弹回调用造成：在滚动结束时，基于超过滚动区域边缘的区域是否在显示区域中可见，而使滚动区域沿与滚动相反的方向弹回。

[10]本发明的至少某些实施例包括具有与软件应用进行交互的用户接口软件的环境。该环境中通过应用编程接口（API）而工作的方法包括传送拉伸调用。当滚动区域超过了显示边缘时，滚动区域在显示区域内拉伸一个预定的最大位移。该方法还包括传送边缘拉伸调用，以设定用于显示器至少一个边缘（例如上下边缘、左右边缘）的位移值。

[11]本发明的至少某些实施例包括具有与软件应用进行交互的用户接口软件的环境，以提供用于装置的显示器的手势操作。用于在该环境中通过用户编程接口（API）进行操作的方法包括传送缩放变换调用。手势操作包括响应于具有两个或更多个输入点的用户输入而执行缩放变换，例如放大或缩小。手势操作还包括响应于具有两个或更多个输入点的用户输入而执行旋转变换以使图像或视图旋转。

[12]本发明的至少某些实施例包括用于给装置的显示器执行动画的方法。该方法包括开始至少一个动画。该方法还包括判定各个动画的进展。该方法还包括基于单一的定时器而完成各个动画。该单一的定时器可以基于显示硬件的重绘间隔。

[13]还描述了执行前述方法中一种或多种的各种装置以及在由处理系统执

行时使处理系统执行这些方法的机器可读介质。

[14]还描述了其他方法、装置和机器可读介质。

附图说明

[15]本发明是通过参照附图来说明的，其中：

[16]图 1 是用于对数据处理装置的用户输入进行响应的方法的流程图；

[17]图 2 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[18]图 3 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[19]图 4 的示意图图示了具有触摸敏感显示器 408 的便携式电子装置 400 的用户接口的一种实施例；

[20]图 5A—图 5C 图示了具有触摸敏感显示器的便携式电子装置 400 的用户接口的至少一些实施例；

[21]图 6A—图 6D 图示了根据一些实施例，将条目的列表滚动至列表终点，在该终点处超过该终点的区域被显示，然后列表被沿相反的方向滚动直到超过该终点的区域不再被显示；

[22]图 7 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[23]图 8 图示了根据本发明的某些教导，用于将装置显示器的滚动锁定在水平或垂直方向的第一和第二滚动角度；

[24]图 9 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[25]图 10 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[26]图 11 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[27]图 12 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[28]图 13 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[29]图 14 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[30]图 15 图示了一种装置的显示器，其具有视图的缩放变换；

[31]图 16A 和图 16B 图示了一种装置的显示器，其带有具有第一和第二缩放因子的视图；

[32]图 17 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[33]图 18 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[34]图 19 是用于对装置的显示器上显示的视图进行动画的方法的流程图；

[35]图 20 是用于对装置的显示器上显示的视图进行动画的方法的流程图；

[36]图 21 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节；

[37]图 22A 和图 22B 图示了对装置的显示器的窗口的调整大小进行同步；

[38]图 23 图示了用于对数据处理装置的显示器上显示的应用的视图的拥有者进行切换的方法；

[39]图 24 图示了用于对装置的显示器上显示的应用的视图进行内存管理的方法；

[40]图 25A 和图 25B 图示了具有层的层级的数据结构，层与视图相关联；

[41]图 26 图示了用于将用户接口的媒体和非媒体内容进行合成以在装置上显示的方法；

[42]图 27 图示了数据结构或层次树，其具有层的层级；

[43]图 28 是根据本发明一种实施例的装置的立体图；

[44]图 29 是根据本发明一种实施例的装置的立体图；

[45]图 30 是根据本发明一种实施例的无线装置的立体图；

[46]图 31 是可以实施根据本发明的实施例的系统的框图；

[47]图 32 示出了根据本发明一种实施例的装置的另一示例；

[48]图 33A 是根据本发明一种实施例处于第一构造（例如在膝上型构造中）的装置的立体图；

[49]图 33B 是根据本发明的一种实施例处于第二构造（例如过渡构造）的图 33A 的装置的立体图；

[50]图 33C 是根据本发明一种实施例处于第三构造（例如平板构造）的图 33A 的装置的立体图；

[51]图 34 以流程图的形式图示了根据本发明某些教导的应用编程接口的细节。

具体实施方式

[52]下面将参照下述细节对本发明的各种实施例和各个方面进行说明，附图将图示各种实施例。下面的说明和附图是对本发明的示例性说明，而不应看作对本发明的限制。描述了大量具体细节以提供对本发明各种实施例的详尽理解。但是在某些情况下，将不对公知的或传统的细节进行描述，以对本发明的实施例提供简要的说明。

[53]下文具体实施方式的某些部分是以算法的形式展现的，这些算法包括对于计算机存储器内储存的数据进行的操作。算法大体上是指造成所需结果的操作的自洽序列。这些操作通常需要或涉及到物理操纵或物理量。通常（但不是必然），这些量采取电信号或磁信号的形式，这些信号能够被储存、传输、合并、比较和以其他方式受到操纵。业已证明，有时（主要是为了通常使用的原因），将这些信号称为位、值、元素、符号、字符、项、数等等是方便的。

[54]但是应当牢记，这些以及类似的所有术语是与适当的物理量相关联的，并且仅仅是应用于这些量的方便的标号。除非下文中明显地以其他形式另有说明外，整个说明书中使用诸如“处理”或“计算”或“判定”或“显示”等术语所进行的说明可以指数据处理系统或类似电子装置进行的动作和处理，所述动作和处理操纵计算机的寄存器和存储器内以物理（电

子)量表示的数据并将其转换成该系统的存储器或寄存器(或其他这类信息存储、传输或显示的装置)内类似地以物理量的形式表示的其他数据。

[55]本发明可以涉及用于执行本申请中所述操作中的一项或多项操作的设备。该设备可以为所需的目的而专门构造,或者也可以包括通用计算机,所述通用计算机由储存在该计算机中的计算机程序选择性地激活(activate)或重构(reconfigure)。这样的计算机程序可以被储存在机器(例如计算机)可读介质中或者存储在适于储存电子指令并分别耦联到总线的任何类型介质中,所述计算机可读介质包括但不限于任何类型的盘(包括软盘、光盘、CD-ROM、和磁光盘)、只读存储器(ROM)、随机存取存储器(RAM)、可擦除可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)、闪存、磁性卡片或光学卡片。

[56]机器可读介质包括用于以由机器(例如计算机)可读的形式储存或传输信息的任何机构。例如,机器可读介质包括只读存储器(ROM);随机存取存储器(RAM)、磁盘储存介质、光学储存介质、闪存装置、以电的、光的、声的或其他的形式传播的信号(例如载波、红外信号、数字信号等)等。

[57]本发明的至少某些实施例包括下述环境中一个或多个应用编程接口:该环境具有与软件应用进行交互的用户接口软件。各种函数调用(function call)或消息通过应用编程接口而在用户接口软件与软件应用之间进行传输。传输这些函数调用或消息可以包括发起(issue)、产生(initiate)、引起(involve)或接收这些函数调用或消息。示例性应用编程接口传输函数调用,用于给具有显示区域的装置实现滚动、手势和动画操作。API还可以实现具有参数、变量或指针的函数。API可以接收所公开的参数或其他参数组合。除了所公开的API之外,其他API也可以单独地或组合地执行与所公开的API类似的功能。

[58]显示区域是窗口的形式。窗口是这样的显示区域:其可以不具有边界,并可以是显示器的整个显示区域或范围。在某些实施例中,显示区域可以具有至少一个窗口和/或至少一个视图(例如网页、文本或图像内容)。窗口可以具有至少一个视图。所公开的方法、系统和设备可以实现

为具有显示区域、窗口和/或视图。

[59]本发明的至少某些实施例包括用于对装置的显示器进行滚动的滚动操作。滚动操作包括当滚动完成时使滚动区域沿与滚动相反的方向弹回（bounce）、当滚动区域超出了显示边界时将滚动区域拉伸（rubberband）一段预定的最大位移、以及设定将滚动锁定在水平或垂直方向的滚动角度。

[60]本发明的至少某些实施例包括用于装置显示器的手势操作。手势操作包括响应于具有两个或更多个输入点的用户输入而执行缩放变换（例如放大或缩小）。手势操作还包括响应于具有两个或更多个输入点的用户输入而执行旋转变换来使图像或视图旋转。

[61]本发明的至少某些实施例包括用于给装置的显示器执行动画的方法。该方法包括启动至少一个动画。该方法还包括确定每个动画的进展（progress）。该方法还包括根据单一的定时器完成每个动画。该单一的定时器可以基于显示硬件的重绘间隔。

[62]本发明的至少某些实施例可以是数字媒体播放器（例如便携式音乐和/或视频媒体播放器）的一部分，该数字媒体播放器可以包括展现媒体的媒体处理系统、储存媒体的存储装置，并且还可以包括与天线系统和媒体处理系统耦联的射频（RF）收发器（例如用于蜂窝电话的 RF 收发器）。在某些实施例中，储存在远程储存装置上的媒体可以通过 RF 收发器传输到媒体播放器。媒体例如可以是音乐或其他音频、静止图像或运动图像中的一者或多者。

[63]便携式媒体播放器可以包括媒体选择装置，例如来自 Cupertino, CA 的 Apple Computer, Inc.的 iPod®或 iPod Nano®上的点击轮输入装置、触摸屏输入装置、按钮装置、可动指点输入装置或其他输入装置。媒体选择装置可以用来对储存在储存装置和/或远程储存装置上的媒体进行选择。至少在某些实施例中，便携式媒体播放器可以包括显示装置，该显示装置耦联到媒体处理系统以显示被通过输入装置选择并被展现的媒体的标题或其他标志物，所述展现是通过扬声器或（一个或多个）耳机进行、在该显示装置上进行或既在该显示装置上又通过扬声器或（一个或多个）耳机进行

的。在某些实施例中，显示装置和输入装置是一体的，而在其他实施例中，显示装置和输入装置是分开的装置。题为“Automated Response to and Sensing of User Activity in Device”的美国专利申请 No.11/586,862 以及美国专利申请公开 No.2003/0095096 和 No.2004/0224638 中描述了便携式媒体播放器的示例，这些文献通过引用方式结合在本申请中。

[64]这里所述发明的实施例可以是其他类型数据处理系统的一部分，这些数据处理系统例如娱乐系统或个人数字助理（PDA）、通用计算机系统、专用计算机系统、嵌入其他装置中的嵌入式装置、不包含媒体播放器的蜂窝电话、多点触摸平板装置、其他多点触摸装置或者将这些装置的各方面或功能相结合的装置（例如将 PDA、娱乐系统和蜂窝电话结合在一个装置中的媒体播放器，例如 iPod®）。在本发明中，电子装置和消费装置是装置的典型示例。

[65]图 1 是用于对装置的用户输入进行响应的方法的流程图。该方法 100 包括在方框 102 处接收用户输入。用户输入的形式可以是输入键、按钮、轮、触摸或用于与该装置进行交互的其他方式。该方法 100 还包括在方框 104 处响应于该用户输入而创建事件对象。该方法 100 还包括在方框 106 处判定该事件对象是否引起滚动或手势操作。例如，在装置的显示器上拖动一端距离的单一触摸可以被解释为滚动操作。在一种实施例中，两个或更多个手指对显示器的触摸可以被解释为手势操作。在某些实施例中，判定事件对象是否引起滚动或手势操作是根据对于给定的时间长度对拖动的接收而进行的。该方法 100 还包括在方框 108 处基于引起滚动或手势操作而发起至少一个滚动或手势调用。该方法 100 还包括：在方框 110 处，基于停在预定位置处的滚动相对于用户输入的滚动量，通过使具有与该事件对象相关的视图（例如网页、文本或图像内容）的窗口滚动，来响应至少一个滚动调用（如果已发起）。滚动可以持续到相对于从用户接收到的最后的输入达到预定位置为止。方法 100 还包括：在方框 112 处，基于以用户输入的形式对多个输入点的接收，通过改变与该事件对象相关的视图，来响应至少一个手势调用（如果已发起）。

[66]在本发明的某些实施例中，滚动操作包括将滚动指示器附着到显示器

的内容边缘。或者，滚动指示器可以被附着到显示边缘。在某些实施例中，鼠标/手指按下的形式的用户输入使滚动指示器显示在滚动区域的显示边缘、内容边缘或窗口边缘处。如果随后检测到鼠标/手指抬起，则滚动指示器从滚动区域的显示边缘、内容边缘或窗口边缘淡出。

[67]在本发明的某些实施例中，手势操作包括基于以用户输入的形式对多个输入点的接收，通过使与事件对象相关的视图旋转，来响应至少一个手势调用（如果已发起）。手势操作可以包括基于对用户输入的接收，通过放大或缩小来对与事件对象相关的视图进行缩放。

[68]在某些实施例中，装置包括具有多个视图或窗口的显示区域。每个窗口可以具有包括主视图（*superview*）和子视图（*subview*）在内的多个视图。必须确定哪个窗口、视图、主视图或子视图被鼠标抬起、鼠标按下或拖动等形式的用户输入所接触。API 可以设定用于进行这种判定的各种模式。在一种实施例中，“通过”模式（*pass mode*）向最接近的子视图发送鼠标按下、鼠标抬起和拖动输入。在另一种实施例中，“拖动时截获”模式（*intercept on drag mode*）在鼠标抬起和按下的输入被发送到子视图的同时向主视图发送拖动输入。在另一种实施例中，“截获”模式（*intercept mode*）向主视图发送全部的拖动、鼠标抬起和按下输入。主视图可以是作为视图软件子集（*subclass*）而工作的滚动器（*scroller*）软件。子视图可以是作为用户接口软件子集而工作的视图软件。

[69]根据本发明的某些教导，图 2 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中以提供弹回操作：该环境具有与软件应用进行交互的用户接口软件。用于提供弹回操作的方法 200 包括在方框 202 处传送设定弹回（*set bounce*）调用。方法 200 还包括在方框 204 处设定最大和最小弹回值中的至少一者。最大和最小弹回值可以与已接收了用户输入的窗口的至少一个边缘相关联。方法 200 还包括：在方框 206 处，在滚动结束时，基于超过滚动区域边缘的区域是否在显示区域中可见，而使滚动区域沿与滚动相反的方向弹回。滚动区域可以是内容区域。

[70]在本发明的某些实施例中，发送固定弹回调用可以是下列任一者：发起、产生、引起或接收该固定弹回调用。

[71]根据本发明的某些教导，图 3 以流程图的形式图示了应用编程接口的细节。该引用编程接口工作于下述环境中以提供拉伸操作：该环境具有与软件应用进行交互的用户接口软件。用于提供拉伸操作的方法 300 包括在方框 302 处传送拉伸调用，以造成对显示器内显示的滚动区域进行拉伸。方法 300 还包括在方框 304 处传送边缘拉伸调用以给显示器的至少一个边缘设定位移值。在某些实施例中，给上下边缘、左右边缘或全部边缘设定位移值。

[72]基于滚动，当滚动区域超过了装置显示器的显示边缘时，根据方法 300 对滚动区域进行的拉伸进行预定的最大位移值。如果用户对显示器的内容进行滚动，使得超过该内容的边缘的区域在显示器中可见，则该位移值对该内容外部的区域的最大量进行限制。在滚动结束时，该内容滑动回去，使得该内容外部的区域不再在显示器上可见。

[73]在本发明的某些实施例中，传输拉伸调用是下列任一者：发起、产生、引起或接收该拉伸调用。

[74]图 4 的示意图图示了具有触摸敏感显示器 408 的便携式电子装置 400 的用户接口的一种实施例。显示器 408 可以包括窗口 410。窗口 410 可以包括一个或多个显示对象，例如信息对象 412-1 至 412-4。在一种示例性实施例中，信息对象 412 可以对应于条目列表中的一个或多个个体的联系人（contact）信息。显示对象可以响应于对与显示器的接触点的运动 414（例如与用户的一个或多个手指 416 相关联，图 4 中的手指没有按比例绘制）进行的检测或判定而运动。在某些实施例中，响应于对接触点的加速运动进行的检测或判定，显示对象的运动可以被加速。尽管实施例 400 包括一个窗口 410，但是在其他实施例中可以有两个或更多个显示窗口。另外，尽管实施例 400 图示了沿特定方向的运动 414，但是在其他实施例中，显示对象的运动可以响应于沿一个或多个其他方向的运动 414 或者响应于一个标量（即被确定的或被检测的与方向无关的运动）。

[75]图 5A—图 5C 图示了将装置上条目的列表滚动到列表的终点，在该位置，列表末端处的一个或多个显示条目平滑地从显示器的末端弹回，翻转方向，然后可选地停止。图 5A 的示意图图示了具有触摸敏感显示器的便

携式电子装置 400 的用户接口的一种实施例。一个或多个显示对象（例如信息对象 412-1）可以与条目列表的终点 514（是滚动区域的边缘）有一段距离 512-1，并可以在该列表被滚动的时候以速度 510-1 运动。注意，终点 514 是与显示对象相关的虚拟边界，这和与窗口 410 和/或显示器 408 相关的物理边界不同。如图 5B 所示，当一个或多个显示对象（例如信息对象 412-1）到达终点 514 或与之相交时，与滚动相对应的运动可以停止，即，滚动速度可以在瞬时为零。如图 5C 所示，这一个或多个显示对象（例如信息 412-1）可以随后翻转方向。在与终点 514 相交之后，信息对象 412-1 可以具有速度 510-2 并可以离终点 514 一段距离 512-2。在某些实施例中，当距离 512-2 等于距离 512-1 时，速度 510-2 的大小可以小于速度 510-1 的大小，即在滚动列表到达其终点并在该处“弹回”之后，所述一个或多个显示对象的运动受到阻尼（damp）。

[76]在本发明的至少某些实施例中，方法 200 执行图 5A—图 5C 所述的弹回操作。方框 202 处传送的弹回调用判定是否允许（enable）弹回操作。最大和最小弹回值确定滚动区域沿与滚动相反的方向的弹回量。

[77]图 6A—图 6D 图示了根据本发明的某些实施例，将条目列表滚动到列表的终点，该终点处超过该终点的区域被显示，随后该列表被沿相反方向滚动，直到超过该终点的区域不再被显示。图 6A—图 6D 的示例中以电子邮件消息作为所列条目来图示了方法 300 的拉伸操作。图 6A—图 6D 图示了根据某些实施例，用于对收件箱（inbox）进行管理的示例性用户接口 3500A。类似的用户接口可以用来显示和管理其他邮箱（例如草稿的、已发送的、垃圾的、个人的等）。另外，也可以有其他类型的列表，包括但不限于即时消息会话、常用电话号码、联系人信息、标签、电子邮件文件夹、电子邮件地址、实际地址、铃声或相册名称的列表。

[78]如果电子邮件的列表填充了比所分配的屏幕区域更多的地方，则用户可以在触摸屏上用垂直向上和/或垂直向下刷（swipe）的手势对这些电子邮件进行滚动。在图 6A 的示例中，电子邮件列表的一部分被显示在屏幕区域中，包括显示在顶部的来自 Bruce Walker 的电子邮件 3530 以及显示在底部的来自 Kim Brook 的电子邮件 3532。用户执行垂直向下的刷手势

3514 以向列表的顶部滚动。垂直向下的手势 3514 不需要精确地垂直；大体上垂直的手势就足够了。在某些实施例中，与精确垂直相差预定角度范围内的手势造成垂直滚动。

[79]作为对垂直向下的手势 3514 进行检测的结果，在图 6B 中，所显示的电子邮件已经向下移位，使得此前显示在底部的来自 Kim Brook 的电子邮件 3532 不再被显示，此前显示在顶部的来自 Bruce Walker 的电子邮件 3530 现在成为从顶部数的第二个，而图 6A 中未显示的、来自 Aaron Jones 的电子邮件 3534 现在被显示在列表的顶部。

[80]在该示例中，来自 Aaron Jones 的电子邮件 3534 是列表中的第一个电子邮件，因此是列表的终点。在到达该电子邮件 3534 时，响应于对垂直向下的手势 3514 的持续检测，该第一个电子邮件 3534 上方（即超过了该列表的终点）的区域 3536（图 6C）被显示。在某些实施例中，超过了列表终点的被显示的区域与列表的背景在视觉上截然不同。在图 6C 中，区域 3536 以及电子邮件（即电子邮件 3534 和 3530）的背景都是白色，因此在视觉上截然不同。

[81]一旦完成了垂直向下的手势 3514，使得触摸屏上或其附近不再检测到相应的对象，则列表被沿相反的方向滚动，直到该区域 3536 不再被显示。图 6D 图示了沿相反方向的这种滚动的结果，现在来自 Aaron Jones 的电子邮件 3534 被显示在分配给该列表的屏幕区域的顶部，而区域 3536 不被显示。

[82]在图 6A—图 6D 的示例中，垂直向下的手势使得列表中第一个条目之外的区域被显示。如图 3 所示，在方框 304 对于上下边缘，或者在方框 306 处对窗口的全部边缘，设定用于预定最大位移的值（例如对列表中第一个条目之外的区域进行显示）。

[83]类似地，如果在列表已经被滚动到最后一个条目时垂直向上的手指在继续，则垂直向上的手势可以使得列表中最后一个条目之外的区域被显示。与第一个条目类似，该最后一个条目可以被看作是列表的终点。如上所述，手势不需要精确垂直以造成垂直滚动；与完全垂直相差预定角度范围内的手势就足够了。

[84]根据本发明的某些教导，图 7 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中以提供方向性滚动操作：该环境具有与软件应用进行交互的用户接口软件。用于通过应用编程接口（API）工作的方法 700 包括在方框 702 传送方向性滚动角度调用，以判定是否允许方向性滚动。方法 700 还包括在方框 704 处传送方向性滚动角度调用，以设定滚动角度，所述滚动角度用于将滚动锁定在垂直方向或水平方向中至少一者。该方法 700 还包括在方框 706 处，如果用户输入与水平方向形成的角度小于或等于第一滚动角度，则将滚动锁定在水平方向。该方法 700 还包括在方框 708 处，如果用户输入与垂直方向形成的角度小于或等于第二滚动角度，则将滚动锁定在垂直方向。

[85]在某些实施例中，以拖动形式进行的用户输入与水平方向形成的角度小于第一滚动角度。在此情况下，用户可能希望沿水平方向滚动。滚动将被锁定在水平方向，直到用户输入超过了第一滚动角度。可以用第二滚动角度来将用户输入锁定在垂直方向。第二滚动角度可以被设定成等于第一滚动角度。

[86]图 8 图示了用于将装置显示器的滚动锁定在水平或垂直方向的第一和第二滚动角度。水平方向 802 和垂直方向 804 是相对于装置的窗口或显示器而言的。如方法 700 中所述，与水平方向 802 形成的角度小于或等于第一滚动角度 806 或 808 的用户输入（例如拖动运动）将把用户输入锁定在水平方向。类似地，与垂直方向 804 形成的角度小于或等于第二滚动角度 810 或 812 的用户输入将把用户输入锁定在垂直方向。第一和第二滚动角度可以设定成相同角度，也可以设定在不同角度。例如，第一和第二滚动角度可以被设定在 25 度。相对于水平或垂直方向小于或等于 25 度的用户输入将把滚动锁定在合适的方向。

[87]在某些实施例中，水平和垂直锁定角度可以部分地由内容的朝向（aspect）来确定。例如，瘦高页面（tall page）形式的内容可以接收比水平锁定角度更大的垂直锁定角度。

[88]根据本发明的某些教导，图 9 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中以提供减速滚动操作：该环境具

有与软件应用进行交互的用户接口软件。用于提供减速滚动操作的方法 900 包括在方框 902 处传送减速滚动调用以设定用于拖动用户输入的减速因子。该方法 900 还包括在方框 904 处根据该拖动用户输入的速度以及该减速因子来将滚动减慢直到停止。

[89]在某些实施例中，拖动形式的用户输入引起特定时间长度的滚动操作。用户输入具有特定速度。通过将减速因子应用于拖动运动期间用户输入的速度，在用户输入停止之后，装置显示器的窗口或显示区域的滚动区域将停止滚动。

[90]根据本发明的某些教导，图 10 以流程图的形式图示了应用编程接口的细节。该用户编程接口工作于下述环境中以提供滚动滞后（hysteresis）操作：该环境具有与软件应用进行交互的用户接口软件。用于提供滚动滞后操作的方法 1000 包括在方框 1002 处传送滚动滞后调用以判定用户输入是否引起滚动。该方法 1000 还包括在方框 1004 处设定用于对用户输入是否引起滚动进行判定的滞后值。

[91]在某些实施例中，在装置的显示器或显示器内的窗口上拖动特定距离的形式的用户输入引起滚动操作。滞后值确定了该特定距离，这个距离是在引起滚动操作之前，用户输入必须在显示器或窗口上拖动的距离。没有拖动该特定距离的用户输入将不会引起滚动操作，并且可以被认为是鼠标抬起或按下输入，或者其他类型的输入。

[92]根据本发明的某些教导，图 11 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中以将滚动指示器附着到装置的滚动区域边缘或窗口边缘：该环境具有与软件应用进行交互的用户接口软件。在某些实施例中，滚动区域边缘与内容边缘相关联。窗口或显示边缘可以与显示区域的边缘相关联。用于提供滚动指示器的方法 1100 包括在方框 1102 处传送滚动指示器调用以判定是否有至少一个滚动指示器附着到滚动区域的边缘或窗口边缘。滚动指示器可以被显示在任何显示边缘、窗口边缘或滚动区域边缘上。该方法 1100 还包括在方框 1104 处根据该滚动指示器调用来可选地将至少一个滚动指示器附着到滚动区域的边缘。或者，该方法 1100 还可以包括在方框 1106 处根据该滚动指示器调用来可选

地将至少一个滚动指示器附着到视图的窗口边缘。

[93]在某些实施例中，方法 1100 的这些操作可以被交替、更改、组合或删除。例如，方框 1104 可以被删除。类似地，方框 1106 可以从方法 1100 中删去。或者，方框 1104 和方框 1106 的次序可以被交换。具有本发明中已经讨论过的各种操作的其他方法也可以被交替、更改、重排、分解、组合或删除。

[94]在本发明的某些实施例中，传送滚动指示器调用是下列任一者：发起、产生、引起或接收滚动指示器调用。例如，用户接口软件（例如软件包或库）可以从软件应用接收滚动指示器调用。

[95]根据本发明的某些教导，图 12 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中以判定是否有无意的用户输入接触到装置显示器的视图：该环境具有与软件应用进行交互的用户接口软件。该方法 1200 包括在方框 1202 处传送无意用户输入调用，以判定用户输入是否是无意的。该方法 1200 还包括在方框 1204 处根据对无意用户输入调用的判定而忽略该无意用户输入。在一种实施例中，无意用户输入调用包括拇指检测调用，以判定用户输入是否是无意的拇指行为。

[96]在本发明的某些实施例中，传送无意用户输入调用是下列任一者：发起、产生、引起或接收无意用户输入调用。

[97]手势 API 在应用与用户软件之间提供接口以处理手势。手势可以包括对视图、窗口或显示器的缩放、旋转或其他变化。掩码（mask）可以只允许特定的改变而限制或不允许其他的改变。所有类型的事件都通过图形构架进入该应用。这些事件被查询、分解（在必要时）、分派（dispatch）。如果这些事件是系统级事件（例如，应用应当被挂起、装置方向已经改变等等），则它们被引导至具有用户接口软件的类的实例（instance）的应用。如果这些事件是基于用户输入的手事件，则这些事件被引导至发生它们的窗口。窗口然后通过对实例的鼠标和手势方法进行调用，来将这些事件引导至合适的控制部分。接收了鼠标按下或鼠标输入函数的控制部分将继续得到将来的全部调用，直到手被抬起。如果检测到第二个手指，则引起手势方法或函数。这些函数可以包括开始、改变和结束手势调用。将来

的全部改变手势调用将被发送到接收了开始手势的控制部分，直到该手势结束。

[98]根据本发明的某些教导，图 13 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中以提供手势操作：该环境具有与软件应用进行交互的用户接口软件。用于提供手势操作的方法 1300 包括在方框 1302 处传送处理手势（handle gesture）事件调用。该方法 1300 还包括在方框 1304 处响应于处理手势事件调用而传送手势变化调用。

[99]在某些实施例中，由装置的显示器以两个或更多个点的形式接收用户输入。装置的多点触摸驱动器接收该用户输入并将该事件包装成事件对象。窗口服务器接收该事件对象并判定该事件对象是否是手势事件对象。如果窗口服务器确定已经接收了手势事件对象，则在方框 1302 处，用户接口软件向与视图相关的软件应用发起或传送该处理手势调用。该软件应用确认已经接收到手势事件并将处理手势调用传递给用户接口软件的库。窗口服务器还将该手势事件对象与接收了该用户输入的视图相关联。在方框 1304 处，库通过响应于处理手势事件调用来传送手势变化调用，从而作出响应。

[100]在一种实施例中，与用户输入相关联的窗口或视图接收这种变化以执行手势事件。提供该视图的用户软件接收手势开始事件调用、手势变化事件调用、用于该视图的缩放设定、以及手势结束调用。这些手势调用接收手势事件的输入，该手势事件可以是具有类型（例如手事件、键盘事件、系统事件等）的基础（base）事件。与该应用相关联的委托（delegate）接收开始手势调用、手势已变化调用、以及手势已完成调用。在手势处理的运行时间期间，用户软件被动态地链接到该应用中。

[101]在某些实施例中，手势变化函数调用包含与手势有关的下列信息：

- 当前按下的手指的数目；
- 最初按下的手指的数目；
- 手的旋转；
- 手的缩放；
- 手的平移；

内侧和最外的手指的位置；以及
第一个手指的按压。

在某些实施例中，可以如下地包括与各个手指按下有关的更多信息。

手指的阶段（刚刚触摸、完全按压、正在抬起等）；

手指的位置；

手指的接近度（触摸有多猛烈）；

手指的方向（卵形接触部分处于什么角度）；

长轴和短轴的长度；

手指的速度；以及

手指的卵形接触部分的偏心率（eccentricity）。

[102]手势事件对象可以是和弦（chord）事件对象，该和弦事件对象具有和弦数目（例如与视图或显示器接触的手指的数目）、和弦开始事件、和弦变化事件、和弦结束事件。和弦变化事件可以包括缩放变换或旋转变换。

[103]根据本发明的某些教导，图 14 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中以提供装置的显示器的显示区域、窗口或视图的缩放变换：该环境具有与软件应用进行交互的用户接口软件。用于提供缩放变换的方法 1400 包括在方框 1402 处传送缩放变换调用，以确定对于与具有多个输入点的用户输入相关联的视图的缩放变换。该方法 1400 还包括在方框 1404 处传送缩放手势开始调用。该方法 1400 还包括在方框 1406 处传送缩放手势进展调用。该方法 1400 还包括在方框 1408 处传送缩放手势结束调用。

[104]在某些实施例中，两个或更多个输入点（例如手指）的形式的用户输入移近或移远，以引起手势事件，所述手势事件对与该用户输入相关联的视图执行缩放变换。缩放变换包括最小和最大缩放因子。图 15 图示了装置的显示器 1502，其具有视图的缩放变换。视图 1504（例如网页、文本或图像内容）具有第一缩放因子。与该视图 1504 相关联的用户输入（例如两个手指移远）被解释为从视图 1504 向视图 1508 进行放大的手势事件，所述视图 1508 具有的第二缩放因子超过了视图 1516 的最大缩放因子。快速返回（snapback）标志确定了在快速返回与视图 1516 相关联的最

大缩放因子之前，上述放大是否可以超过最大缩放因子而达到视图 1508。

[105]图 16A 图示了装置的显示器 1604，其具有视图 1616 的第一缩放因子。与视图 1614 相关联的用户输入（例如两个手指 1608 和 1610 移近）被解释为从视图 1614 向视图 1664 进行放大的手势事件，所述视图 1664 具有如图 16B 所示的第二缩放因子。虚线区域 1602 和 1650 表示内容的全部区域，此时显示区域 1604 和 1652 中只显示了该内容。在执行从图 16A 向图 16B 的缩放变换时，手势事件的中心（对于图 16A 是中心 1612，对于图 16B 是中心 1660）相对于显示器 1604 保持在相同的位置。在变换过程中，滚动指示器 1606 收缩成滚动指示器 1654，以表明由于放大操作，显示器 1604 上正在显示全部内容 1650 的更少部分。虚线区域 1650 比虚线区域 1602 大，以表明由于放大操作，内容的更多部分未被显示在图 16B 中的显示器 1652 上。

[106]在本发明的至少一些实施例中，如图 16C 所示，用户希望将视图 1670 从缩放因子 2x 变化到缩放因子 1x。第一组用户输入 1672 和 1674 向第二组用户输入 1676 和 1678 移动将使缩放因子从 2x 减小到 1x。用户可能希望从 2x 缩到 1x 而不必将用户输入在视图 1670 上移动很大距离。在具有与软件应用进行交互的用户接口软件的环境中，可以设定手势缩放变换标志以确定对于与具有多个输入点的用户输入相关联的视图进行的缩放变换。该缩放变换标志或者从当前缩放因子缩放到最小缩放因子，或者从当前缩放因子变换到最大缩放因子。例如，可以在与 1.5x 的缩放因子和第三组用户输入 1680、1682 相关联的位置设定标志。如果手势缩放变换标志已经被设定在 1.5x 的缩放因子处，则希望将缩放因子从 2x 变化到 1x 的用户只需将其手指（用户输入）从第一组 1672 和 1674 移动到第三组 1680 和 1682。

[107]根据本发明的某些教导，图 17 以流程图的形式图示了应用编程接口的细节。应用编程接口工作于下述环境中以提供装置的显示器的视图、窗口或显示区域的旋转变换：该环境具有与软件应用进行交互的用户接口软件。用于提供旋转变换的方法 1700 包括在方框 1702 处传送旋转变换调用，以确定对于与具有多个输入点的用户输入相关的视图进行旋转变换。

该方法 1700 还包括在方框 1704 处传送旋转手势开始调用。该方法 1700 还包括在方框 1706 处传送旋转手势进展调用。该方法 1700 还包括在方框 1708 处传送旋转手势结束调用。

[108]在某些实施例中，两个或更多个输入点的形式的用户输入进行旋转以引起手势事件，该手势事件对于与该用户输入相关联的视图执行旋转变换。旋转变换包括最小和最大旋转程度，用于相关的最小和最大旋转视图。在视图快速返回到最大旋转程度之前，用户输入可以暂时地将视图旋转到超过旋转的最大程度。

[109]根据本发明的某些教导，图 18 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中，以向委托通知与装置显示器的显示区域、窗口或视图相关的至少一个动画：该环境具有与软件应用进行交互的用户接口软件。该 API 可以指定动画的延迟。另外，该 API 可以给多个动画赋予优先度。用于通知委托的方法 1800 包括在方框 1802 处确定是否发生了任何动画。该方法 1800 还包括在方框 1804 处检查动画的进展。如果已经发生了进展，则在方框 1806 处可以计算该动画的下一阶段（例如位置、不透明性（opacity）或者变形）。如果在方框 1804 处进展已经完成，则在方框 1808 处判定与该完成的动画相关的视图是否与委托相关联。如果是，则在方框 1810 处传送委托调用，以向该委托通知对于该视图的动画。在该软件应用的控制下工作的委托可以响应于由该动画改动的视图而使其他视图变化。

[110]在某些实施例中，软件引起动画，所述动画对与用户输入相关的视图执行缩放变换。显示器可以包括多个视图。由缩放变换增大了尺寸的视图可能阻挡其他视图，在此情况下，可能需要减小其他视图的尺寸。或者，由缩放变换减小了尺寸的视图可能产生多余的区域供其他视图增大尺寸。

[111]图 19 是用于使装置的显示器上显示的显示区域、窗口或视图进行动画（animate）的方法的流程图。该方法 1900 包括在方框 1902 处开始至少两个动画。该方法 1900 还包括在方框 1904 处确定各个动画的进展。该方法 1900 还包括在方框 1906 处根据单一的定时器而完成各个动画。

[112]在本发明的某些实施例中，所述单一的定时器包括基于重绘间隔的定

时器，所述重绘间隔是装置的显示器显示当前帧与下一帧之间的事件长度。在此情况下，动画的变化在重绘间隔期间被更新至显示器，以显示显示器的下一帧期间的变化。各个动画的进展可以被周期性地计算或者基于进展调用来计算。

[113]该方法 1900 还可以包括判定各个动画是否与委托相关联。然后向该委托通知该动画。不与动画相关联的其他视图可以取决于对该委托进行控制的软件应用而变化。

[114]图 20 是用于对装置的显示器上显示的显示区域、窗口或视图进行动画的方法的流程图。该方法 2000 包括在方框 2002 处提供单一的动画定时器。该方法 2000 还包括在方框 2004 处用该单一的动画定时器对多个动画进行动画。例如，单一的定时器可以控制同时发生的全部动画。这些动画可以包括变形的、帧的以及不透明性的动画。动画变形可以包括缩放变换或旋转变换。帧的动画可以包括帧的调整大小（resize）。不透明性的动画使不透明性从不透明变化到透明，或者反过来。

[115]根据本发明的某些教导，图 21 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于下述环境中，以对与装置的显示器的多个视图或窗口相关的动画进行同步：该环境具有与多个软件应用或进程进行交互的用户接口软件。用于对动画进行同步的方法 2100 包括在方框 2102 处给正与进程相关联的各个视图独立地设定视图的属性。例如，视图的属性或特性可以包括位置、大小、不透明性等等。该方法 2100 还包括在方框 2104 处传送同步调用，从而针对显示器的多个视图对动画进行同步。同步调用可以包括输入参数或变量，例如进程的同步的标识、以及正在请求对多个视图进行动画的那些进程的列表。在一种实施例中，同步调用包括正在请求进行动画的进程的数目和标识。在一种实施例中，每个应用或进程在不同的事件发送同步调用。方法 2100 还包括在方框 2106 处当同步标志被允许时传送同步确认消息。当要被同步的这些进程已经各自向操作该用户接口软件的窗口服务器发送了消息时，同步标志可以被允许。该方法 2100 还包括在方框 2108 处将这些视图的属性独立地从第一状态更新至第二状态。在一种实施例中，窗口服务器在不同的事件从各个进程接收经

更新的属性。该方法 2100 还包括在方框 2110 处，当两个进程都具有与第二状态相关联的经更新的属性时，传送开始动画调用以绘制所请求的动画。

[116]在一些实施例中，第一数据结构或层次树表示与这些进程的视图或窗口相对应的那些层的层级（*hierarchy*）。第二数据结构或渲染树（*render tree*）表示该层次树的类似拷贝。但是，渲染树不被更新，直到这些独立的处理已经完成其各自的动画为止。此时，渲染树用新的动画对屏幕进行更新和重绘。

[117]图 22A 和图 22B 图示了对装置的显示器的各个视图或窗口的调整大小进行同步。例如，通过从第一状态（图 22A 中的窗口 2210）变化到第二状态（图 22B 中的窗口 2210），具有尺寸属性的、与第一进程相关联的窗口 2210 可以增大尺寸。大致同时，第二窗口 2220 可以与第一窗口 2210 的尺寸增大成比例地减小尺寸。该方法 2100 对图 22A 和图 22B 所示的窗口 2210 和 2220 的调整大小提供同步。从第一状态向第二状态变化的动画可以渐增地发生，也可以在方法 2100 进行同步的情况下发生。

[118]图 23 图示了对数据处理装置的显示器上显示的应用的视图的拥有者进行切换的方法。该方法 2300 包括在方框 2302 处构造数据结构，该数据结构具有层的层级，层与视图相关联并拥有该视图。这些层可以是用于该应用的用户接口的内容、窗口、视频、图像、文本、媒体或任何其他类型的对象。该方法 2300 还包括在方框 2304 处从数据结构除去层。该方法 2300 还包括在方框 2306 处将视图的拥有者从层切换到视图。

[119]在某些实施例中，来自该数据结构的每个层与视图相关联。与视图相关联的层向该视图发送委托函数调用，以生成由该视图提供的内容。第一指针引用从该层指向该视图。第二指针引用从该视图指向该层。指向对象（例如该视图）的引用的数目被定义为该对象的保留数（*retained count*）。该视图可以接收将要从该数据结构除去该层的通知。从数据结构除去层可以基于与从装置的显示器除去的层相关联的视图而发生。当从数据结构或层次树除去层时，从该层指向该视图的指针将被除去。该视图将具有为零的保留数，并且在视图的拥有者不被撤销（*reverse*）的情况下从

存储器解除分配 (deallocate) 或除去。如果拥有者被撤销, 则该视图将具有至少为一的保留数。

[120]图 24 图示了在装置的显示器上显示的应用的视图的存储器管理的方法。方法 2400 包括在方框 2402 处构造数据结构, 该数据结构具有层的层级, 至少一个层与该视图相关联。方法 2400 还包括在方框 2404 处在存储器中储存数据结构。方法 2400 还包括在方框 2406 处维持从其他对象到该视图的引用的数目的保留数。方法 2400 还包括在方框 2408 处, 如果保留数为零, 则将视图从存储器解除分配。如上所述, 如果从数据结构除去层, 则视图的保留数将递减。从数据结构除去层可以基于与从装置的显示器除去的层相关联的视图而发生。

[121]图 25A 和图 25B 图示了一种数据结构, 其具有层的层级, 层与视图相关联。该数据结构包括层 2502、2504 和 2506。层 2506 与视图 2510 相关联。与视图 2510 相关联的层 2506 向该视图发送委托调用, 以生成由该视图提供的内容。第一指针引用 2508 从层 2506 指向视图 2510。第二指针引用 2512 从视图 2510 指向层 2506。第三指针引用 2532 可以从用户接口 (UI) 控制器 2530 指向视图 2510。UI 控制器 2530 可以控制与视图 2510 相关的操作, 例如响应于用户输入而滚动视图 2510。基于指针引用 2508 和 2532, 图 25A 中的视图 2510 具有至少为二的保留数。

[122]如果如图 25B 所示从该数据结构除去层 2506, 则指针 2508 被除去。视图 2510 将如图 25B 所示具有更低的保留数。如果视图 2510 具有为零的保留数, 则储存该视图 2510 的存储器将被解除分配。

[123]图 26 图示了用于对用户接口的媒体和非媒体内容进行合成以在装置上显示的方法。该方法 2600 包括在方框 2602 处构造数据结构, 该数据结构具有与装置的用户接口相关联的各层的层级。该方法 2600 还包括在方框 2604 处判定该数据结构的各个层与媒体还是非媒体内容相关联。该数据结构或层次树被遍历 (traverse), 以确定该数据结构的各个层与媒体还是非媒体内容相关联。该方法 2600 还包括在方框 2606 处从该数据结构将与媒体内容相关联的层分离 (detach)。该方法 2600 还包括在方框 2606 处将媒体内容储存在第一存储器位置。该方法 2600 还包括在方框 2608 处

将非媒体内容储存在第二存储器位置。该方法 2600 还包括在方框 2610 处对该媒体和非媒体内容进行合成以在装置上显示。

[124]在一些实施例中，对媒体和非媒体内容进行合成包括从第一存储器位置取回媒体内容、从第二存储器位置取回非媒体内容、并将媒体和非媒体内容直接扫描到显示器。存储器位置可以是位于缓存器、主存储器、图形处理单元或装置内其他位置中的任何类型的存储器。媒体内容可以包括视频、视频插件、音频、图像或其他随事件而变的媒体。媒体内容可以是 YUV 模型的形式，其中 Y 代表亮度成分（明亮度），而 U 和 V 代表色度（色彩）成分。媒体内容可以以每秒大约 20 至 40 帧的速率被扫描到显示器。媒体内容可以在被扫描到装置的显示器之前受到缩放。

[125]非媒体内容可以包括不需要频繁更新的内容、视图和图像。非媒体内容可以是作为加色（additive）模型的 RGB 模型的形式，其中红色、绿色和蓝色（经常用在加色光模型中）被以各种方式组合以产生其他颜色。非媒体内容可以以比媒体内容更低的速率被扫描到显示器。

[126]图 27 图示了数据结构或层次树，其具有层的层级。这些层与媒体和非媒体内容相关联。例如，层 2704 与媒体内容 2706（例如视频）关联。层 2710 与非媒体内容 2712 关联，非媒体内容 2712 可以是用于该视频的用户接口视图。层 2720、2730 和 2740 分别与形成非媒体内容 2712 的各组成部分的非媒体内容 2722、2732 和 2742 相关联。方法 2600 将确定该数据结构的每个层是否与媒体或非媒体内容相关联。与媒体内容相关联的任何层（例如层 2704）将被从该数据结构除去，并在另外的存储器位置受到处理。

[127]在一些实施例中，本发明的方法、系统和设备可以在各种装置中实现，这些装置包括电子装置、消费装置、数据处理装置、台式计算机、便携式计算机、无线装置、蜂窝装置、平板装置（tablet device）、手持装置、多点触摸装置、多点触摸数据处理装置、这些装置的任意组合或者其他类似的装置。图 4—6 和图 28—33 图示了一些这些装置的示例。

[128]图 28 图示了根据本发明一种实施例的装置 2800。图 28 示出了一种具有“糖块”样式的电话构造的无线装置。在图 28 中，无线装置 2800 可以

包括壳体 2832、显示装置 2834、输入装置 2836（可以是字母数字键区（keypad））、扬声器 2838、麦克风 2840 和天线 2842。无线装置 2800 还可以包括接近度传感器 2844 和加速度计 2846。可以理解，图 28 的实施例可以使用更多或更少的传感器，并可以具有与图 28 所示形状因素（form factor）不同的形状因素。

[129]所示的显示装置 2834 位于壳体 2832 的上部，所示输入装置 2836 位于壳体 2832 的下部。所示天线 2842 在壳体 2832 的上部从壳体 2832 延伸。所示扬声器 2838 也在壳体 2832 的上部处于显示装置 2834 上方。所示麦克风 2840 在壳体 2832 的下部处于输入装置 3286 下方。可以理解，扬声器 2838 和麦克风 2840 可以位于壳体上任何位置处，但是通常分别位于与用户的耳朵和嘴巴相对应的位置。

[130]显示装置 2834 例如可以是不具备接收输入的能力的液晶显示器（LCD），也可以是包括 LCD 的触摸输入屏。输入装置 2836 例如可以包括按钮、开关、拨号盘（dial）、滑动件（slider）、键或键区、导航板、触摸板、触摸屏等。可以把任何公知的扬声器、麦克风和天线分别用于扬声器 2838、麦克风 2840 和天线 2842。

[131]从接近度传感器 2844 和加速度计 2846 获取的数据可以被组合或单独使用，以收集与用户的行为有关的信息。来自接近度传感器 2844、加速度计 2846 或这二者的数据例如可以用来激活/关断（activate/deactivate）显示器背光，产生命令，作出选择，对显示器上的滚动、手势、动画或其他运动进行控制，对输入装置设定进行控制，或者对装置的一个或多个设定进行其他变化。在本发明的某些实施例中，装置 2800 可以用来实施本发明中所述方法的至少一部分。

[132]图 29 示出了根据本发明一种实施例的装置 2950。装置 2950 可以包括壳体 2952、显示/输入装置 2954、扬声器 2956、麦克风 2958 和可选的天线 2960（可以在壳体外部可见，也可以隐藏在壳体内部）。装置 2950 还可以包括接近度传感器 2962 和加速度计 2964。装置 2950 可以是蜂窝电话，或者 PDA 与蜂窝电话集成为一体的装置，或者媒体播放器与蜂窝电话集成为一体的装置，或者既是娱乐系统（例如用于玩游戏）又是蜂窝电

话的装置，或者装置 2950 可以是本申请中所述其他类型的装置。在一种具体实施例中，装置 2950 可以包括蜂窝电话、媒体播放器和 PDA，它们都包含在壳体 2952 内。装置 2950 可以具有这样的形状因素：其足够小以适合握在正常成年人的手掌中，并且足够轻以能够由成年人用一只手携带。可以理解，术语“便携”表示该装置能够容易地握持在成年人的手掌（一只或两只）中，例如，膝上型计算机和 iPod 是便携式装置。

[133]在一种实施例中，显示/输入装置 2954 除了作为显示器（例如 LCD）之外，还可以包括多点触摸输入屏。在一种实施例中，多点触摸屏是电容感知介质，构造成对在触摸面板的平面内的不同位置处同时发生的多个触摸（例如从用户的脸在显示器上的接触形状（blob），或者多个手指同时触摸或接近触摸该显示器）或者接近触摸（例如显示器上的接触形状）进行检测并产生不同的信号，这些信号代表触摸面板的平面上对于多个触摸中每个触摸的触摸位置。

[134]在本发明的某些实施例中，装置 2800 可以用于实施本发明中讨论的方法中的至少一部分。

[135]图 30A 和图 30B 图示了根据本发明一种实施例的装置 3070。装置 3070 可以是包括铰链 3087 的蜂窝电话，铰链 3087 将显示器壳体 3089 连接到键区壳体 3091。铰链 3087 使用户能够开启和关闭蜂窝电话，使之可以被置于图 30A 和图 30B 所示两种不同构造中的至少一者。在一种具体实施例中，铰链 3087 可以将显示器壳体以可旋转方式连接到键区壳体。具体而言，用户可以开启蜂窝电话，使之处于图 30A 所示的开启构造，也可以关闭蜂窝电话，使之处于图 30B 所示的关闭构造。键区壳体 3091 可以包括键区 3095 和麦克风 3097，键区 3095 接收来自用户的输入（例如电话号码输入或其他字母数字输入），麦克风 3097 接收来自用户的语音输入。显示器壳体 3089 可以在其内部表面上包括显示器 3093（例如 LCD）、扬声器 3098 和接近度传感器 3084；在其外部表面上，显示器壳体 3089 可以包括扬声器 3096、温度传感器 3094、显示器 3088（例如另一个 LCD）、环境光传感器 3092 以及接近度传感器 3084A。因此，在本实施例中，显示器壳体 3089 可以在其内部表面上包括第一接近度传感器并

在其外部表面上包括第二接近度传感器。

[136]在至少某些实施例中，装置 3070 可以包含提供下列一项或多项功能的部件：无线通信装置（例如蜂窝电话）、媒体播放器、娱乐系统、PDA 或本申请中所述的其他类型装置。在一种实施例的实现形式中，装置 3070 可以是一体地集成了媒体播放器的蜂窝电话，所述媒体播放器播放 MP3 文件，例如 MP3 音乐文件。

[137]图 4、5A、5B、5C、6A、6B、6C、6D、28、29、30A 和 30B 所示的各种装置可以是无线通信装置（例如蜂窝电话），并可以包括提供无线通信能力的多个部件。图 31 示出了无线装置 3070 的一种实施例，其包括用于无线通信的能力。无线装置 3070 可以包括在图 4、5A、5B、5C、6A、6B、6C、6D、28、29、30A 和 30B 所示的任一装置中，不过图 4、5A、5B、5C、6A、6B、6C、6D、28、29、30A 和 30B 的那些装置的替换实施例也可以包括比无线装置 3070 更多或更少的部件。

[138]无线装置 3070 可以包括天线系统 3101。无线装置 3070 还可以包括数字和/或模拟射频（RF）收发器 3102，其耦联到天线系统 3101 以通过天线 3101 发送和/或接收语音、数字数据和/或媒体信号。

[139]无线装置 3070 还可以包括数字处理系统 3103，以控制数字 RF 收发器和管理语音、数字数据和/或媒体信号。数字处理系统 3103 可以是通用处理装置，例如微处理器或控制器。数字处理系统 3103 还可以是专用处理装置，例如 ASIC（专用集成电路）、FPGA（现场可编程门阵列）或 DSP（数字信号处理器）。数字处理系统 3103 还可以包括本领域已知的其他装置以与无线装置 3070 的其他部件进行接口。例如，数字处理系统 3103 可以包括模拟—数字转换器和数字—模拟转换器，以与无线装置 3070 的其他部件进行接口。数字处理系统 3103 可以包括媒体处理系统 3109，媒体处理系统 3109 可以包括通用或专用处理装置以管理媒体，例如音频数据文件。

[140]无线装置 3070 还可以包括存储装置 3104，存储装置 3104 耦联到数字处理系统以储存数据和/或用于无线装置 3070 的操作程序。存储装置 3104 例如可以是任何类型的固态或磁性存储装置。

[141]无线装置 3070 还可以包括一个或多个输入装置 3105，该输入装置耦联到数字处理系统 3103 以接受用户输入（例如电话号码、名称、地质、媒体选择等等）。输入装置 3105 例如可以是下列一项或多项：键区、触摸板、触摸屏、与显示装置结合的指点装置或类似的输入装置。

[142]无线装置 3070 还可以包括至少一个显示装置 3106，显示装置 3106 耦联到数字处理系统 3103 以显示信息，例如消息、电话呼入信息、联系人信息、图片、电影和/或通过输入装置 3105 而选择的媒体的标题或其他标志物。显示装置 3106 例如可以是 LCD 显示装置。在一种实施例中，显示装置 3106 和输入装置 3105 可以一体集成在同一装置中（例如触摸屏 LCD（例如多点触摸输入面板），其与显示装置（例如 LCD 显示装置）一体集成）。显示装置 3106 可以包括背光 3106A，以在特定状况下对显示装置 3106 进行照明。可以理解，无线装置 3070 可以包括多个显示器。

[143]无线装置 3070 还可以包括电池 3107，以向系统的各个部件提供工作电力，这些部件包括：数字 RF 收发器 3102、数字处理系统 3103、存储装置 3104、输入装置 3105、麦克风 3105A、音频换能器 3108、媒体处理系统 3109、（一个或多个）传感器 3110、显示装置 3106。电池 3107 例如可以是可充电的或不可充电的锂离子电池或镍金属氢化物电池。无线装置 3070 还可以包括音频换能器 3108 和至少一个麦克风 3105A，音频换能器 3108 可以包括一个或多个扬声器。在本发明的某些实施例中，无线装置 3070 可以用来实施本发明中所述各种方法的至少一部分。

[144]图 32 示出了根据本发明实施例的装置的另一种示例。该装置 3200 可以包括处理器（例如微处理器 3202）和存储器 3204，处理器和存储器通过总线 3206 彼此耦联。装置 3200 可以可选地包括耦联到微处理器 3202 的缓存器 3208。该装置还可以可选地包括显示控制器和显示装置 3210，其通过总线 3206 耦联到其他部件。一个或多个输入/输出控制器 3212 也可以耦联到总线 3206，以提供用于输入/输出装置 3214 的接口并提供用于一个或多个传感器 3216 的接口，传感器 3216 可以用于感知用户行为。总线 3206 可以包括如本领域所知通过各种桥、控制器和/或适配器而彼此连接的一个或多个总线。输入/输出装置 3214 可以包括键区、键盘或例如触摸

输入面板的光标控制装置。而且，输入/输出装置 3214 可以包括网络接口，网络接口可以用于有线网络或无线网络（例如 RF 收发器）。传感器 3216 可以是本申请中所述那些传感器中的任一种，例如包括接近度传感器或环境光传感器。在装置 3200 的至少某些实施方式中，微处理器 3202 可以接收来自一个或多个传感器 3216 的数据并可以以本申请中所述的方式对该数据执行分析。例如，可以通过人工智能处理或以本申请中所述的其他方式对该数据进行分析。作为分析的结果，微处理器 3202 然后可以自动地使得对装置的一个或多个设定进行调整。

[145]在本发明的某些实施例中，装置 3200 可以用来实施本申请中所述那些方法的至少一部分。

[146]图 33A—图 33C 示出了根据本发明至少某些实施例的装置的另一种示例。图 33A 图示了膝上型装置 3300，其具有键盘 3302、主体 3304、显示器框架 3306 和显示器 3308。膝上型装置 3300 可以被转变成图 33B 和图 33C 所示的平板装置。图 33B 图示了该膝上型装置向平板装置的转变。包含显示器 3358 的显示器框架 3356 的边缘在主体 3354 内滑过键盘 3352 的顶部，直到形成如图 33C 所示的平板装置。该平板装置的显示器 3362 和显示器框架 3366 置于主体 3360 的顶上。

[147]在本发明的某些实施例中，膝上型装置 3300 可以用来实施本申请中所述那些方法的至少一部分。

[148]根据本发明的某些教导，图 34 以流程图的形式图示了应用编程接口的细节。该应用编程接口工作于具有与软件应用进行交互的用户接口软件的环境中。在一些实施例中，视图的层级工作于用户接口软件内的层的层级之上。API 以如方法 3400 所示的方式工作，该方法包括在方框 3402 处构造视图的层级，该层级工作于层的层级之上。该方法 3400 还包括在方框 3404 处提供对于视图层级的访问（access）而不提供对于层的层级的访问。应用可以通过该 API 与视图的层级进行交互，而不访问在视图层级之下工作的层的层级。

[149]在某些实施例中，平台提供各种滚动、手势和动画操作。该平台包括硬件部件和操作系统。硬件部件可以包括耦联到输入面板的处理单元以及

耦联到处理器的存储器。操作系统包括一个或多个程序，所述程序储存在存储器中并构造成由处理单元执行。一个或多个程序包括各种指令，用于通过应用编程接口来传送函数调用或消息，以执行各种滚动、手势和动画操作。

[150]在一种实施例中，这一个或多个程序包括用于通过 API 传送缩放变换调用的指令，以确定用于与用户输入相关的视图的缩放变换，所述用户输入具有多个输入点。在一种实施例中，这一个或多个程序包括用于通过 API 传送拉伸调用的指令，以确定与用户输入相关联的视图的旋转变换，所述用户输入具有多个输入点。

[151]在一些实施例中，该平台包括包含软件代码库的构架（framework）。该构架与平台的程序进行交互，以提供应用编程接口，用于执行各种滚动、手势和动画操作。该构架还包括储存在单个目录中的相关的资源（例如图像、文本等）。

[152]在一种实施例中，该构架的库提供 API 以给与具有多个输入点的用户输入相关联的视图指定缩放变换。在一种实施例中，该构架的库提供 API 以给与具有多个输入点的用户输入相关联的视图指定缩放变换。

[153]在前述说明中，已经参照本发明的具体示例性实施例对本发明进行了说明。显然，在不脱离由所附权利要求阐述的发明精神和范围的情况下，可以对其进行各种修改。因此，说明书和附图应当认为是举例说明性的，而不是限制性的。

附录 A API 软件代码

09PI1289

```

- (AffineTransform)transform;
- (void)setTransform:(AffineTransform)transform: // animatable

@interface UIView(Animation)

+ (void)beginAnimation: (String*)animationIDS; // animationID can be nil if you
don't want to have an identifier for the group
+ (void)beginAnimations: (String*)animationID context: (void*) context // additional
context info passed to will start/did stop selectors
+(void)endAnimations;

+(void)setAnimationDelegate:(id)delegate;
+(void)setAnimationDuration:(TimeInterval)duration;
+(void)setAnimationDelay:(TimeInterval)delay;
+(void)setAnimationStartTime:(AbsoluteTime)start;
+(void)setAnimationCurve:(UIViewAnimationCurve)curve;

- (BOOL)handleEvent: (Event)event;

// this is called if the app is going to be suspending back to a previous
application.
// the app should return YES if it is going to perform animation, otherwise it should
// return NO. The startTime argument is a start time for any animations to begin so that
// the animations are synchronized with a window server. If the application's
// implementation of this method returns YES, it should later call [UIApp
suspendWith ANimation:NO] when
// it is finished animating.
(BOOL)animateSuspensionReturningToLastApp:(AbsoluteTime)startTime;

// Set the status bar mode, orientation over duration. If processSynchID is non-zero, used as
sync ID.
- (void) setStatusBarMode: (UIStatusBarMode) mode
orientation: (UIStatusBarOrientation) orientation duration:(float)duration
processSynchID: (int)processSynchID;
// Sets the status bar mode. 0 duration means no animation
- (void) setStatusBarMode: (UIStatusBarMode)mode
orientation: (UIStatusBarOrientation)orientation duration: (float)duration;
- (void) setStatusBarMode: (UIStatusBarMode)mode duration:(float)duration; //
uses the default orientation.
- (UIStatusBarOrientation) statusBarOrientation;
- (UIStatusBarOrientation) statusBarMode;
// Status bar rect in non-rotated window coordinates.

- (void) synchronizedDrawingWithID: (int)processSynchID; // used for status bar
resize
// _____

```

```

// Gesture delegate method. event will be nil if gesture was not event-driven (ie
programmatically or animation).
@interface Object(UITableViewGestureDelegate)

- (void)willStartGestures: (UIView *)view forEvent: (GSEvent)event;

- (void)didFinishGestures: (UIView *)view forEvent: (Event)event;

- (void)willStartGesture: (UIGestureRecognizer)gesture inView: (UIView *) view
forEvent: (Event)event;

- (void)gestureDidChange: (UIGestureRecognizer)gesture inView: (UIView *) view
forEvent: (Event)event;

- (void)didFinishGesture: (UIGestureRecognizer)gesture inView: (UIView *) view
forEvent: (Event)event;
@end

UIGestureRecognizerDelegateZooms // Number (BOOL)

typedef enum {
    UIGestureRecognizerDelegateModeInterceptOnDrag, // send mouseDown / mouseUp to
subviews if not dragged
    UIGestureRecognizerDelegateModePass, // send
mouseDown/mouseDrag/mouseUp to subviews
    UIGestureRecognizerDelegateModeIntercept, // send mouseDown / mouseUp to subviews
} UIGestureRecognizerDelegateMode;
- (void)setEventMode:(UIGestureRecognizerDelegateMode)mode;

- (void)setDirectionalScrolling:(BOOL)flag;

- (void)setDirectionalScrollingAngle:(float)angle;

- (void)setScrollDecelerationFactor:(float)factor;

- (void)setBounces:(BOOL)bounces;

- (void)setThumbDetectionEnabled:(BOOL)flag;

- (void)setShowScrollerIndicators:(BOOL)flag;

- (void)setScrollerIndicator:(CGRect)rect;

- (void)setScrollerIndicatorsPinToContent:(BOOL)flag;

- (void)setAllowsRubberBanding:(BOOL)flag; // CallsetRubberBand:forEdges: with
RUBBER_BAND_HEIGHT for the top and bottom edges

- (void)setAllowsFourWayRubberBanding:(BOOL)flag; // Calls

```

setRubberBand: for Edges: with RUBBER_BAND_HEIGHT for all edges

- (void)setRubberBand: (float) value for Edges: (UIRectEdge) edges: // Default set for top and bottom edges

- (void)contentMouseUpInView:(UIView*)mouseUpView
withEvent:(Event)event; // default calls -mouseUp; on view

- (Point)scroller:(UIScroller *)scroller
adjustSmoothScrollEnd:(Point)offset velocity:(Size)velocity;

@interfaceobject (UIAnimationDelegate)

-(void)animator: (UIAnimator*)animator
startAnimation:(UIAnimation*)animation;

-(void)animator: (UIAnimator*)animator
stopAnimation:(UIAnimation*)animation;

-(void)animator: (UIAnimator*)animator stopAnimation: (UIAnimation*)animation
fraction: (float) fraction;

@end

//

```
@interface UITransformAnimation : UIAnimation {
    @protected
    AffineTransform    _startTransform;
    AffineTransform    _endTransform;
```

}

-(void)setStartTransform: (AffineTransform)start;

-(void)setEndTransform: (AffineTransform)end;

- (AffineTransform)transformForFraction: (float) fraction;

@end

//

```
@interface UIFrameAnimation : UIAnimation {
    Rect    _startFrame;
    Rect    _endFrame;
    UIRectFields    _fieldsToChange;
```

}

- (void) setStartFrame: (Rect)start;

- (void) setEndFrame: (Rect)end;

- (void) setSignificantRectFields: (UIRectFields)fields;

-(Rect)endFrame;

@end

```
@interface UIAlphaAnimation : UIAnimation {
    @protected
    float    _startAlpha;
    float    _endAlpha;
```

}

- (void)setStartAlpha: (float)start;

```
- (void)setEndAlpha: (float)end;  
- (void)alphaForFraction: (float)fraction;
```

```
@end
```

```
@interface UIRotationAnimation : UIAnimation {  
    float _startAngle;  
    float _endAngle;  
- (void)setStartRotationAngle: (float)degrees;  
- (void)setEndRotationAngle: (float) degrees;
```

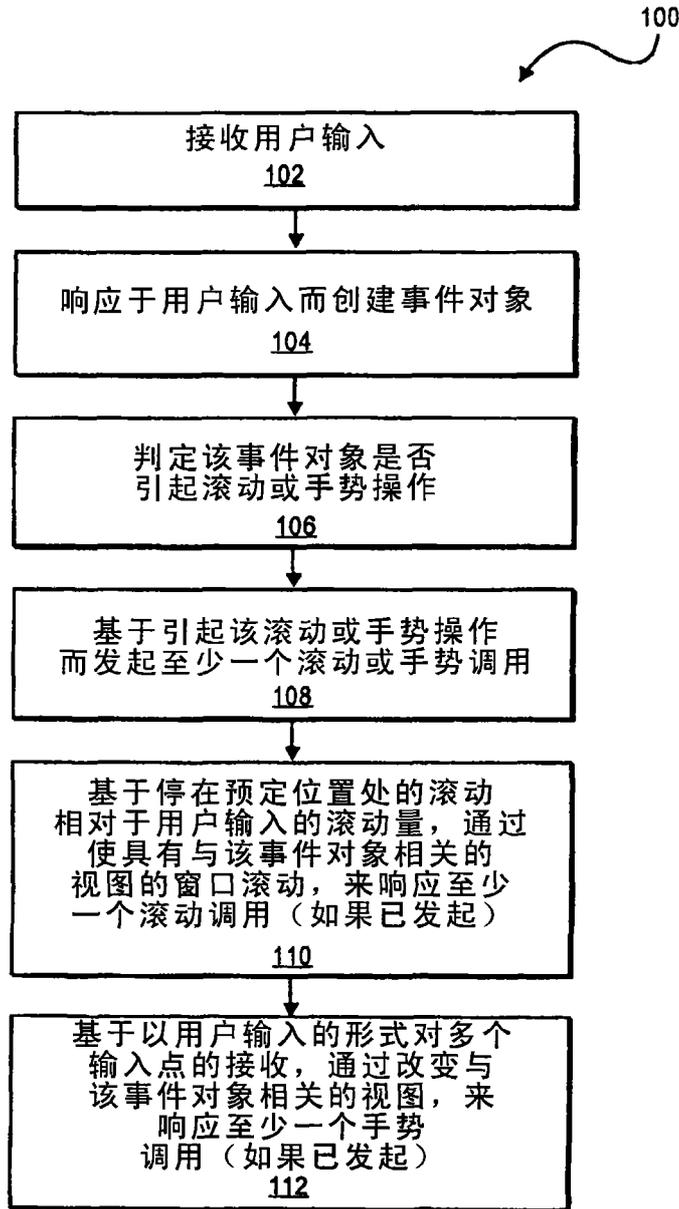


图1

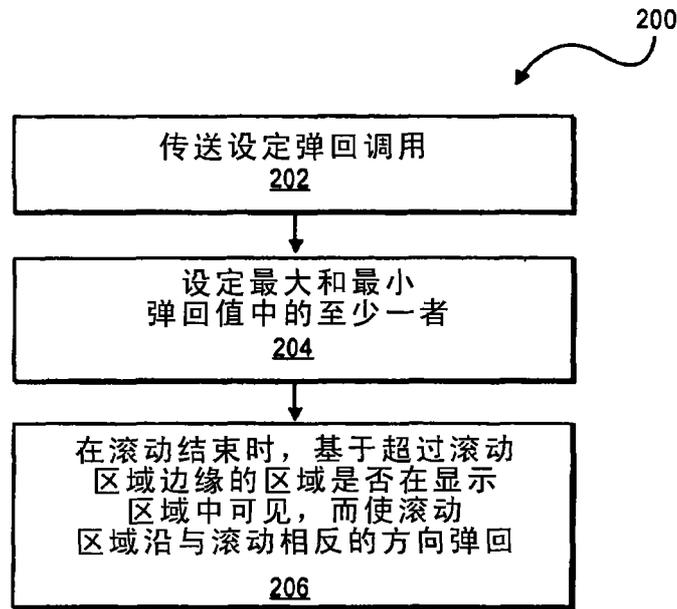


图2

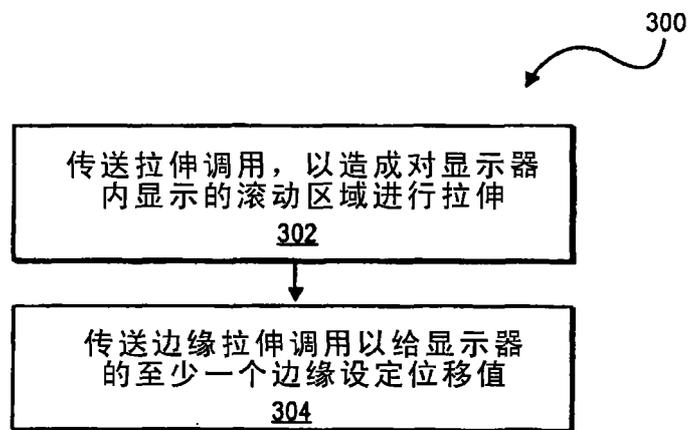


图3

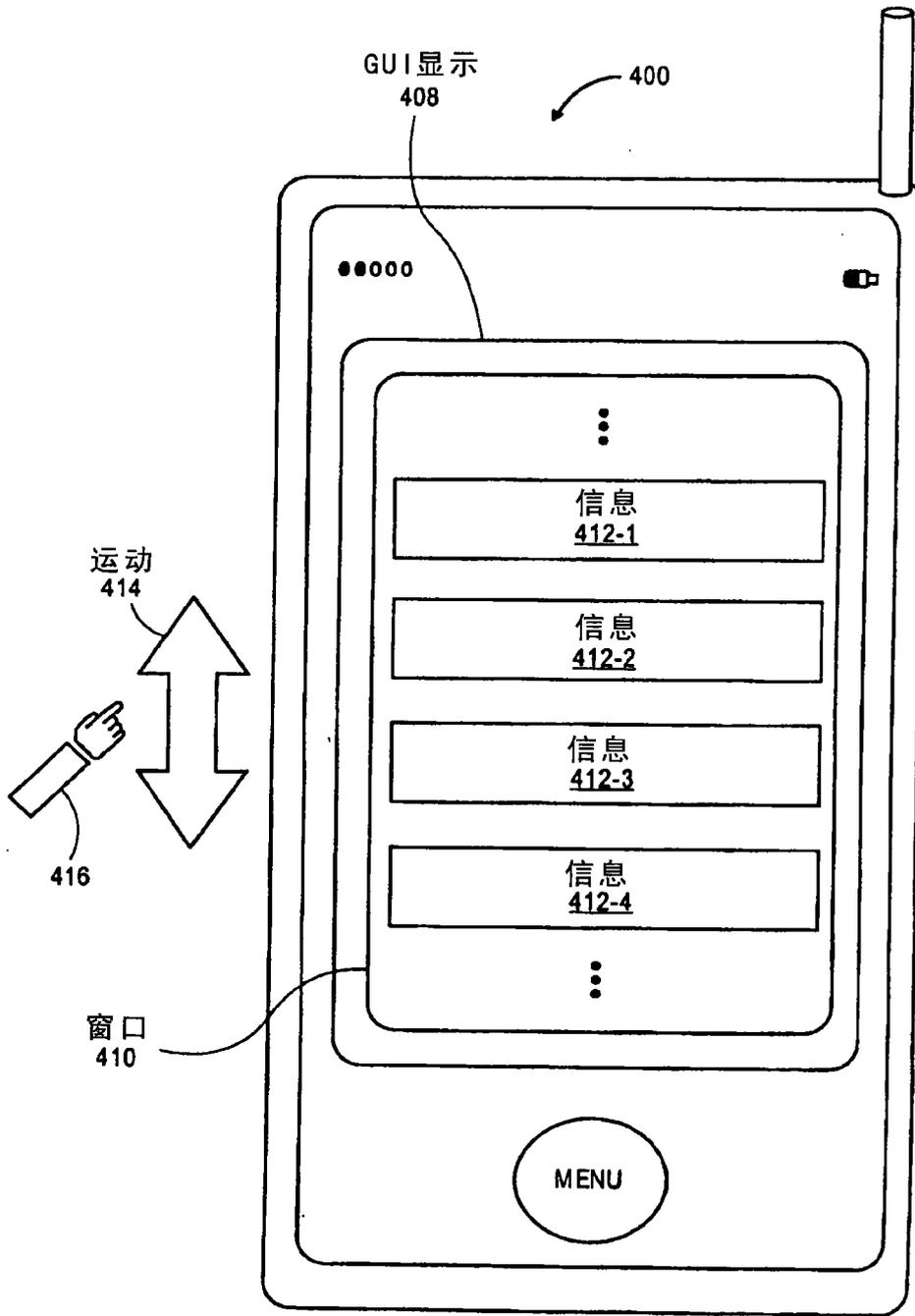


图4

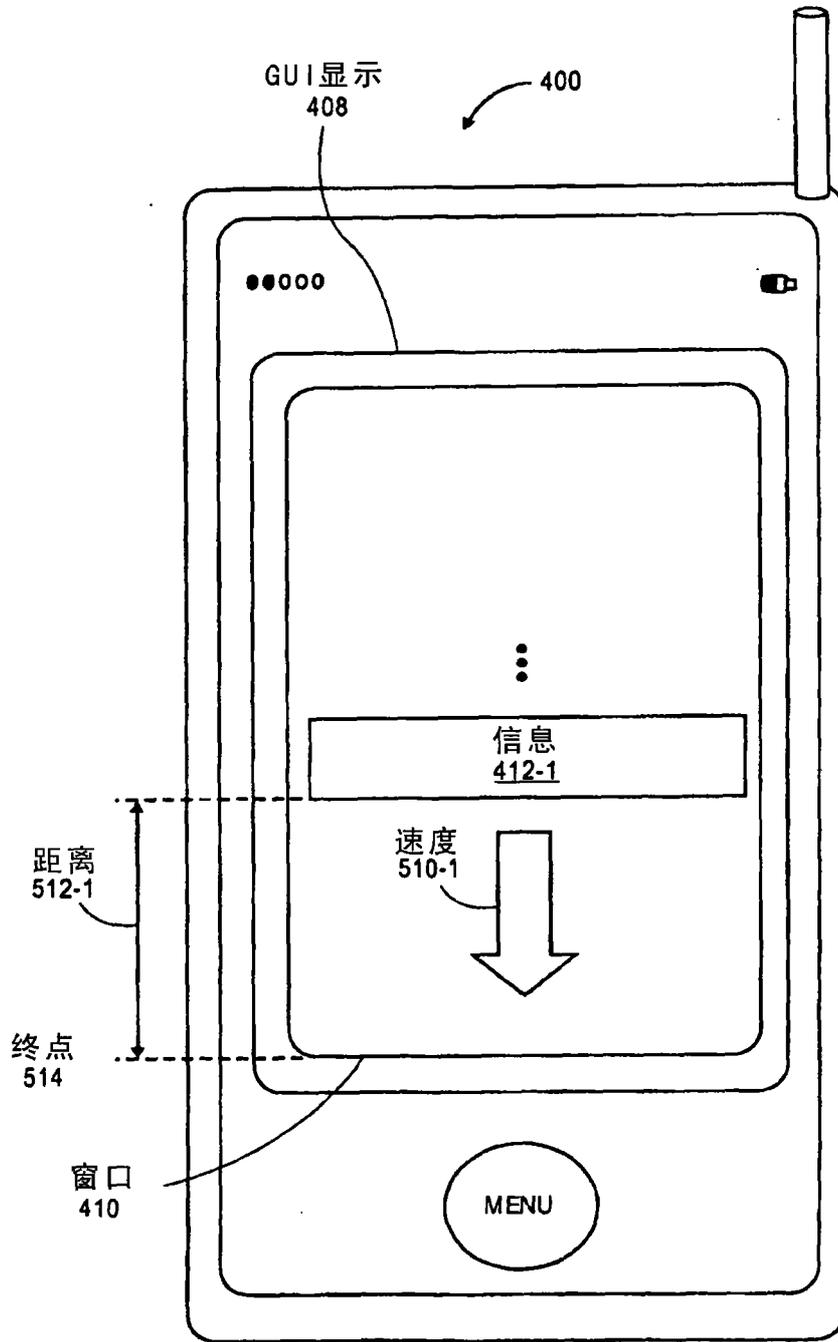


图5A

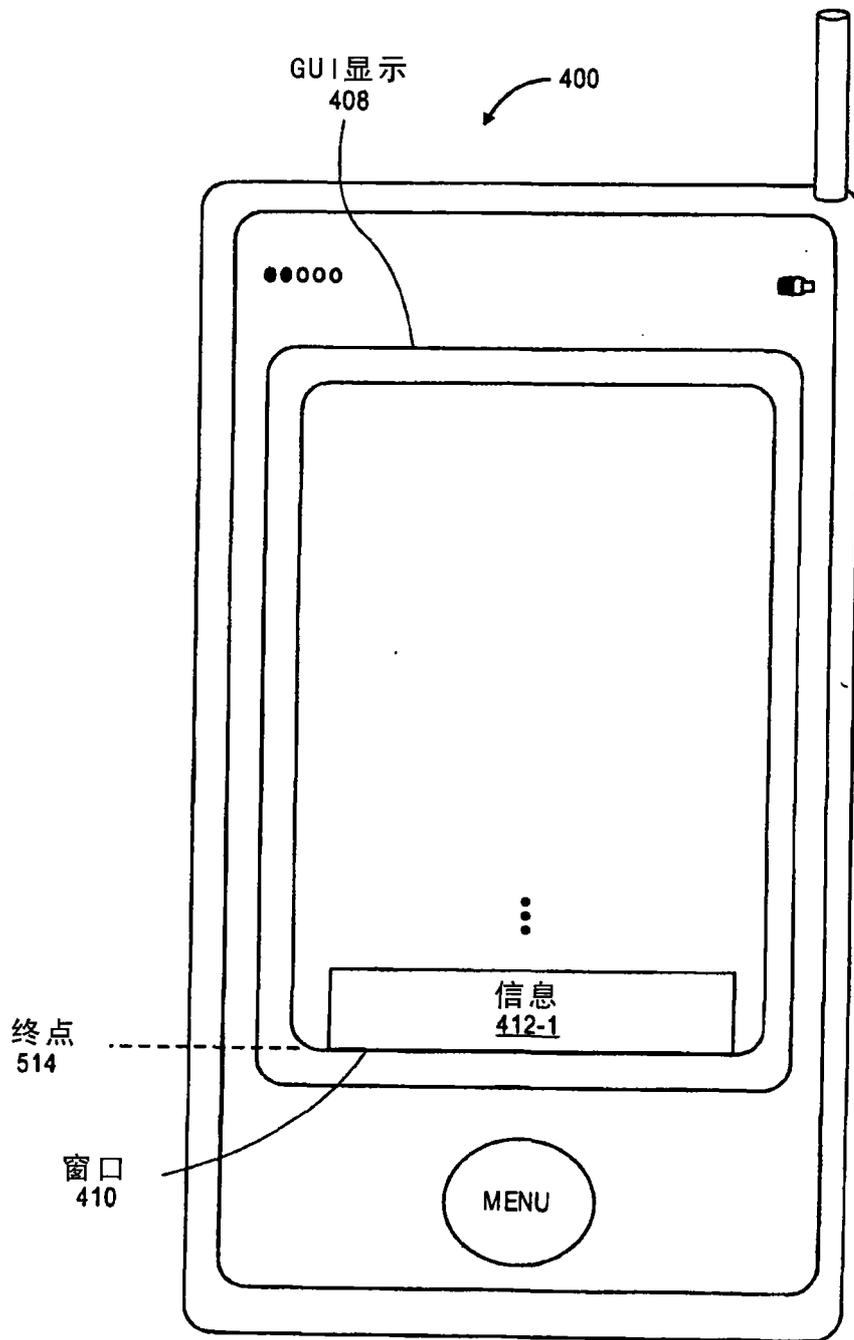


图5B

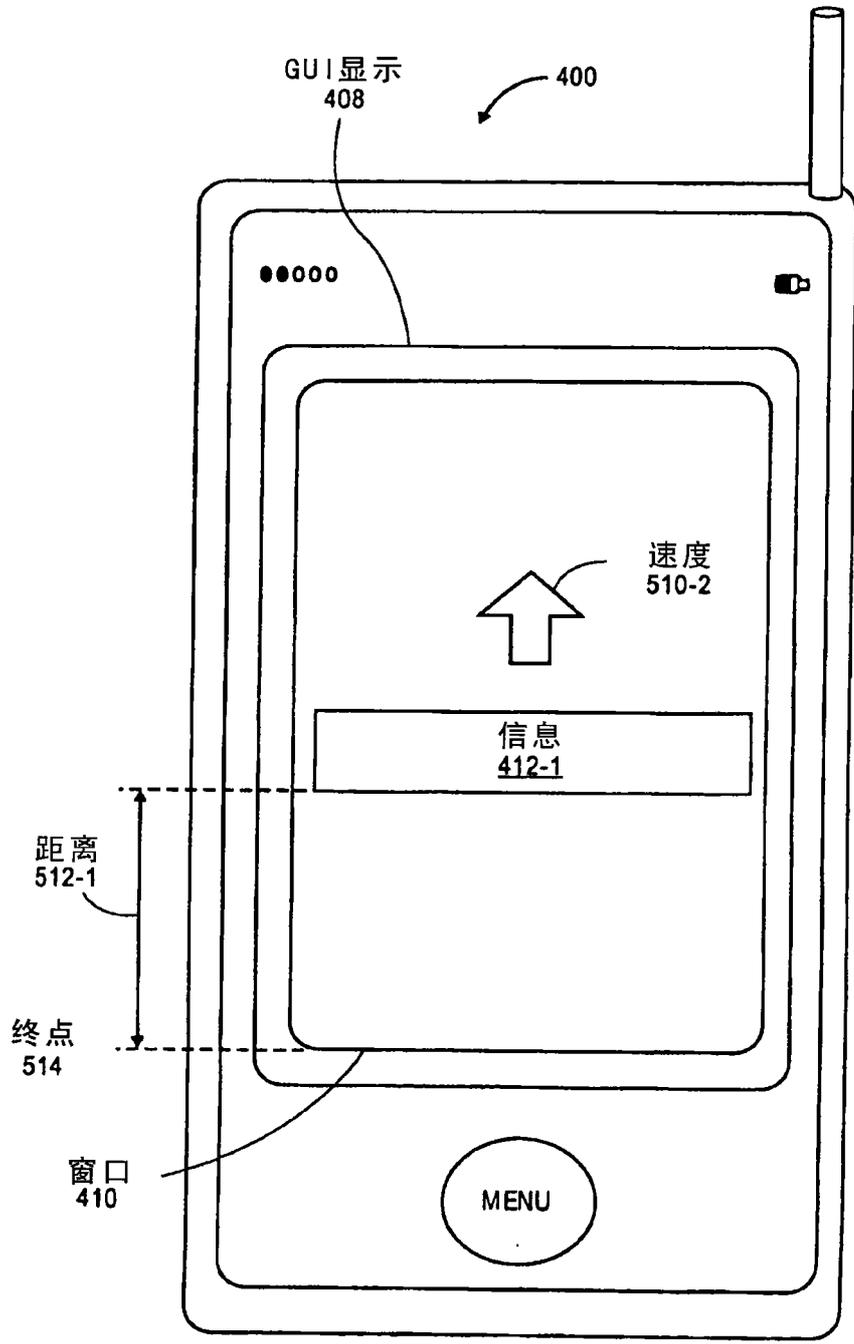


图5C

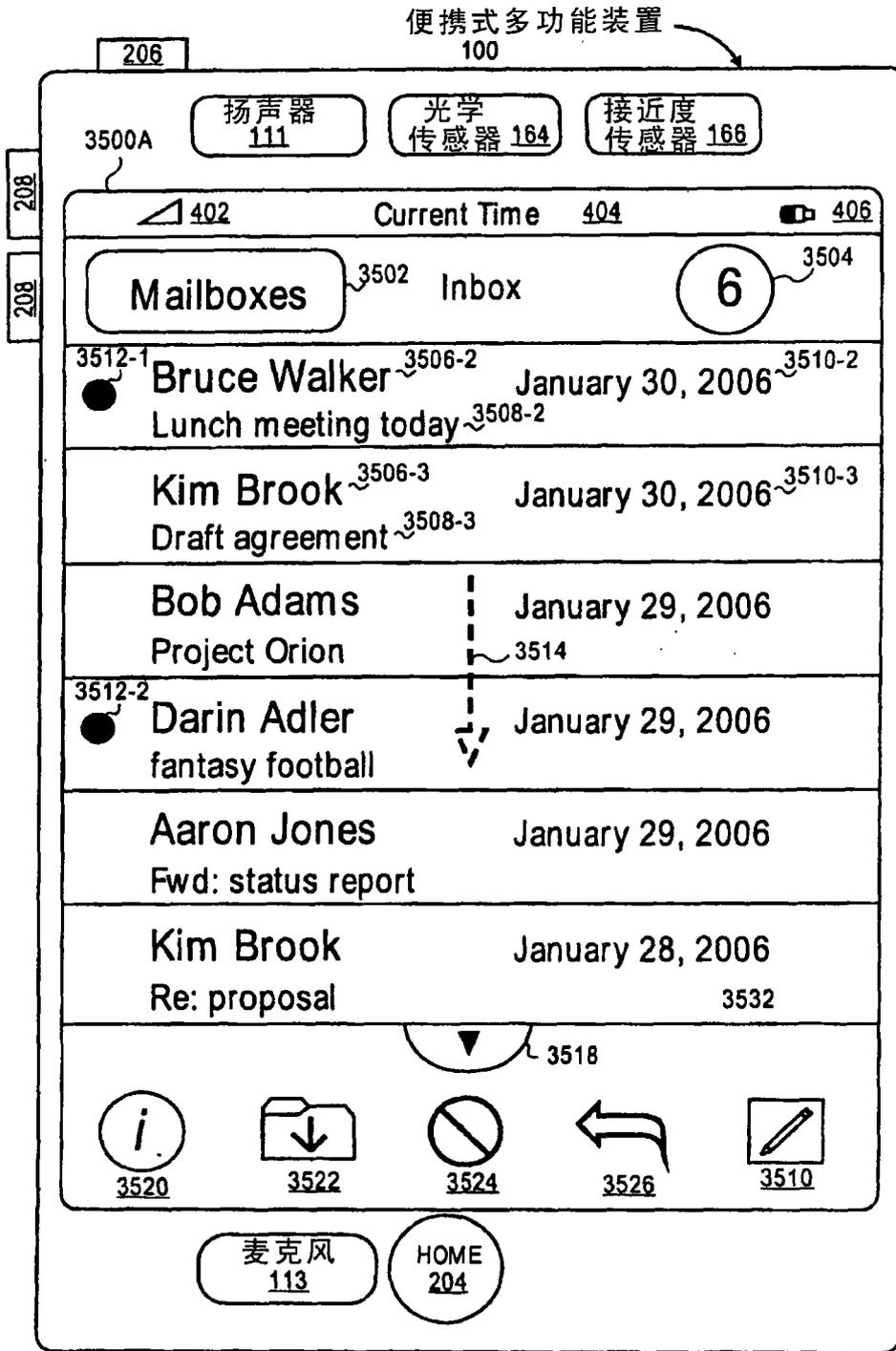


图6A

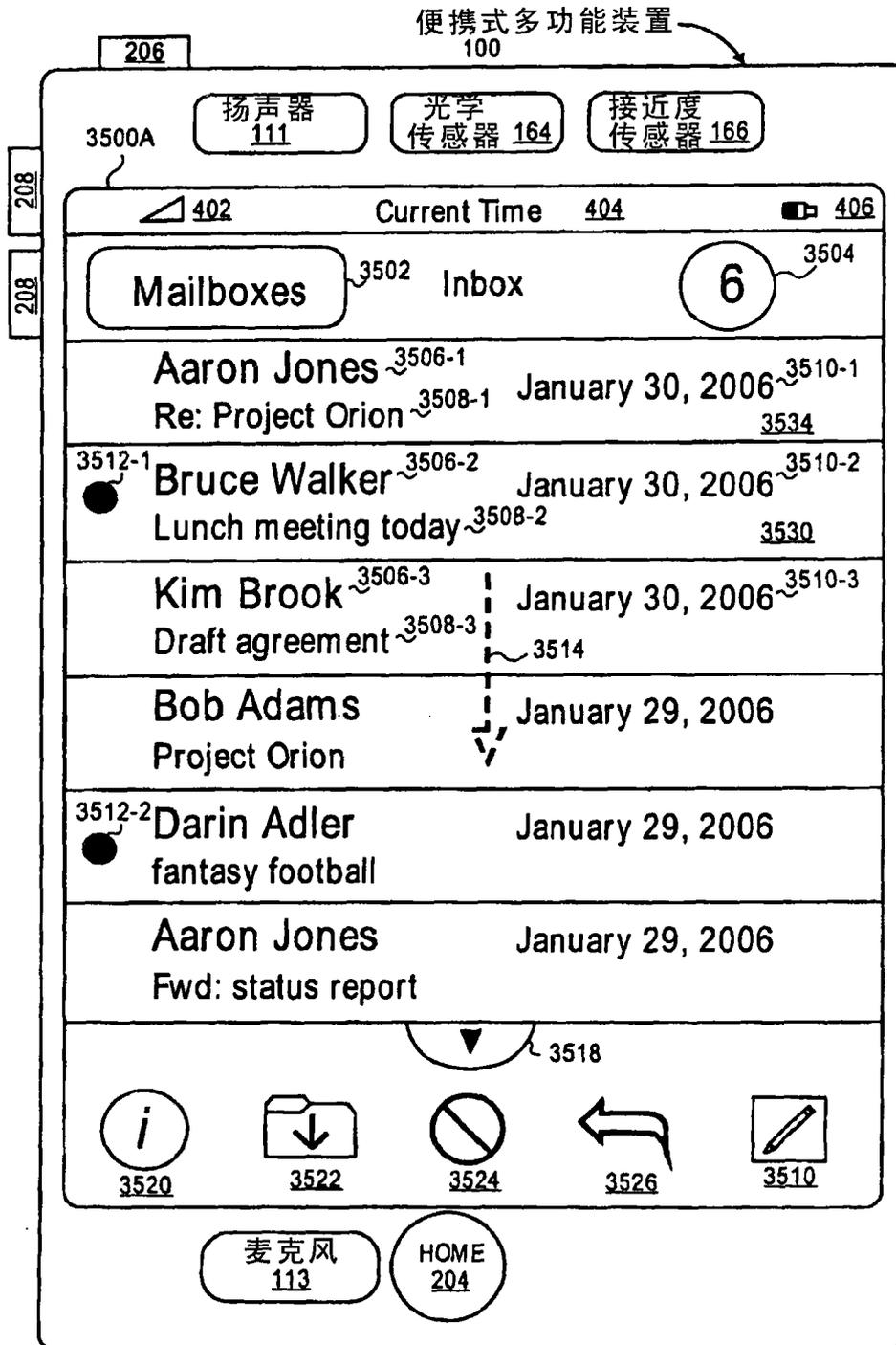


图6B

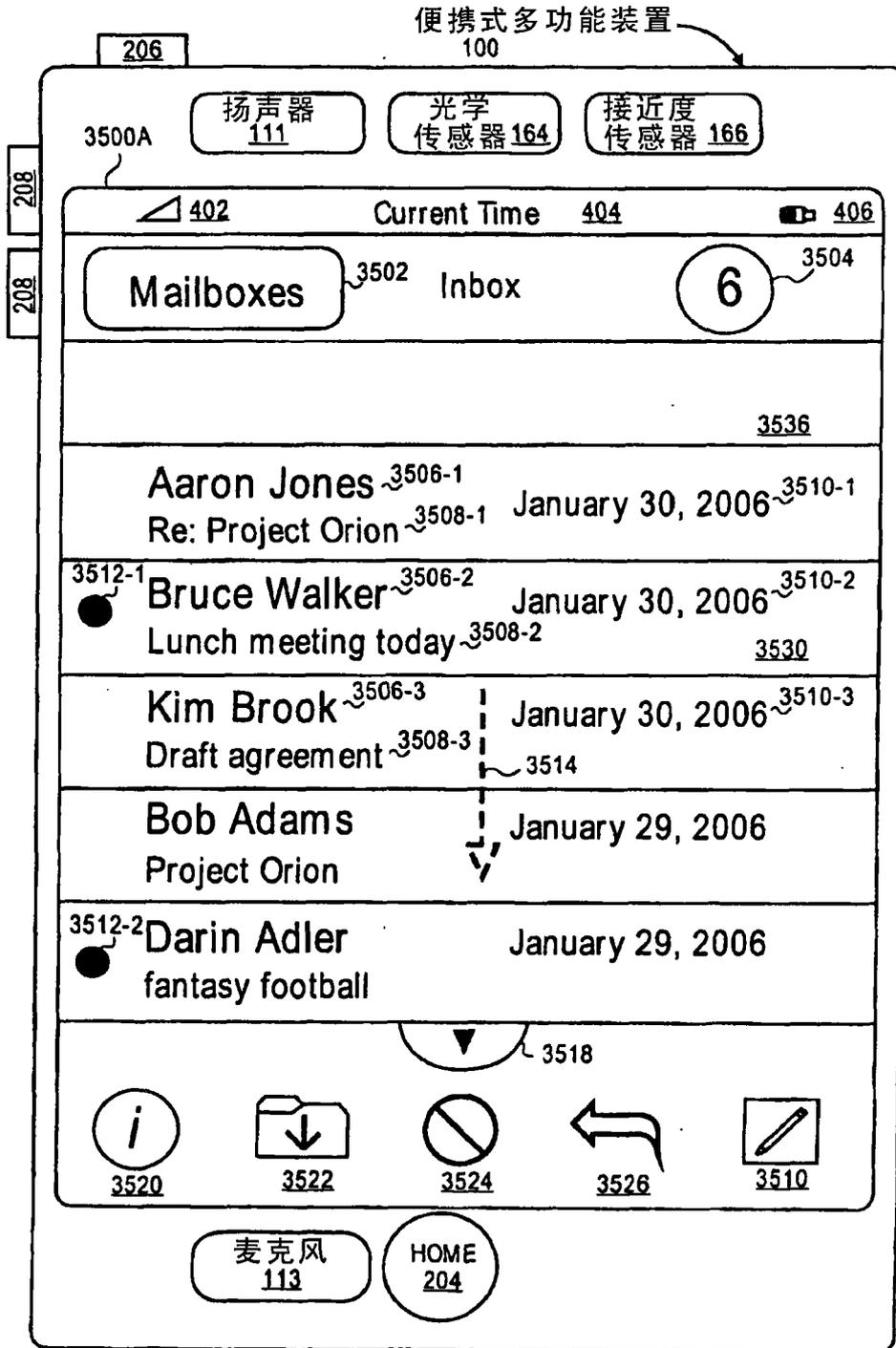


图 6C

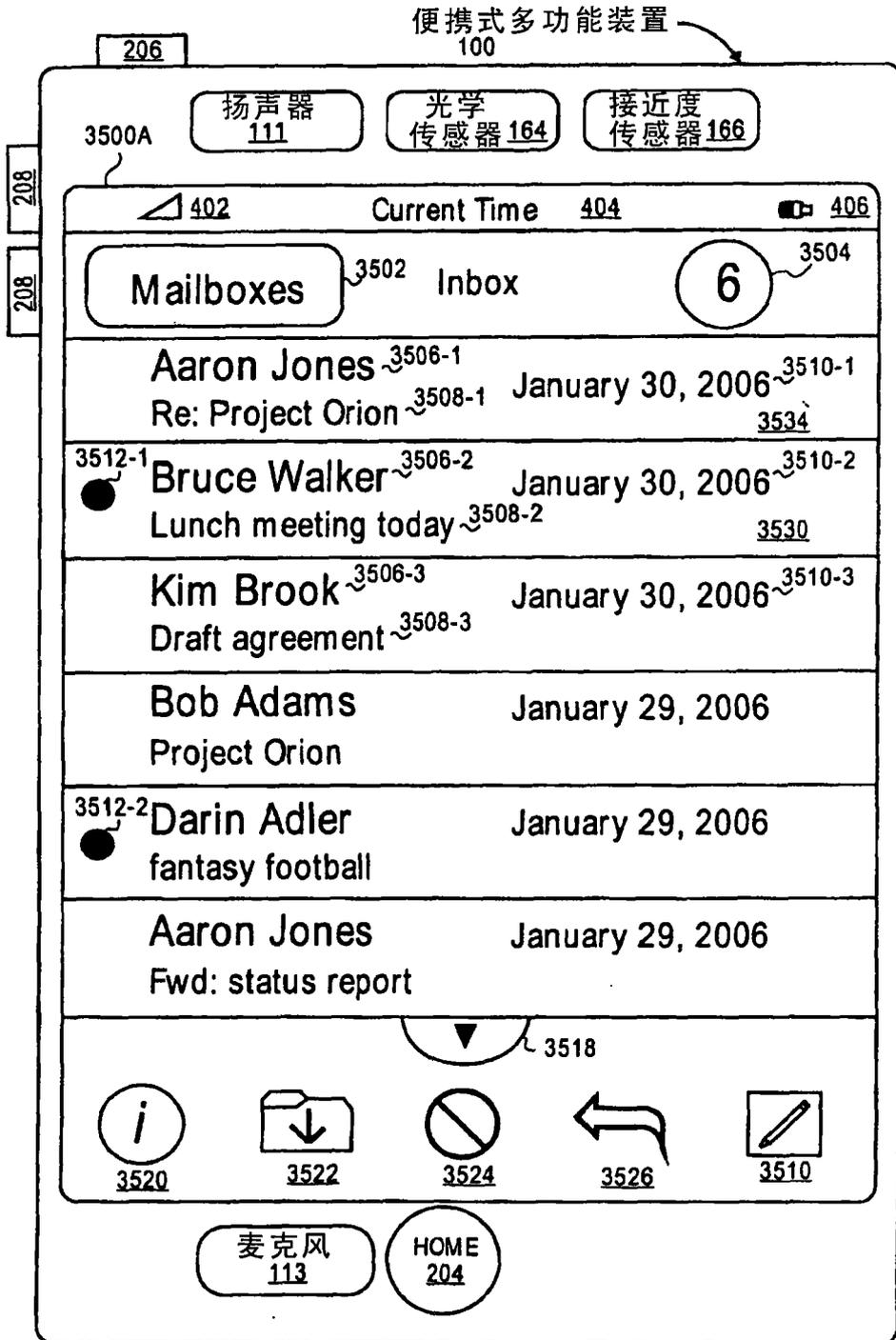


图6D

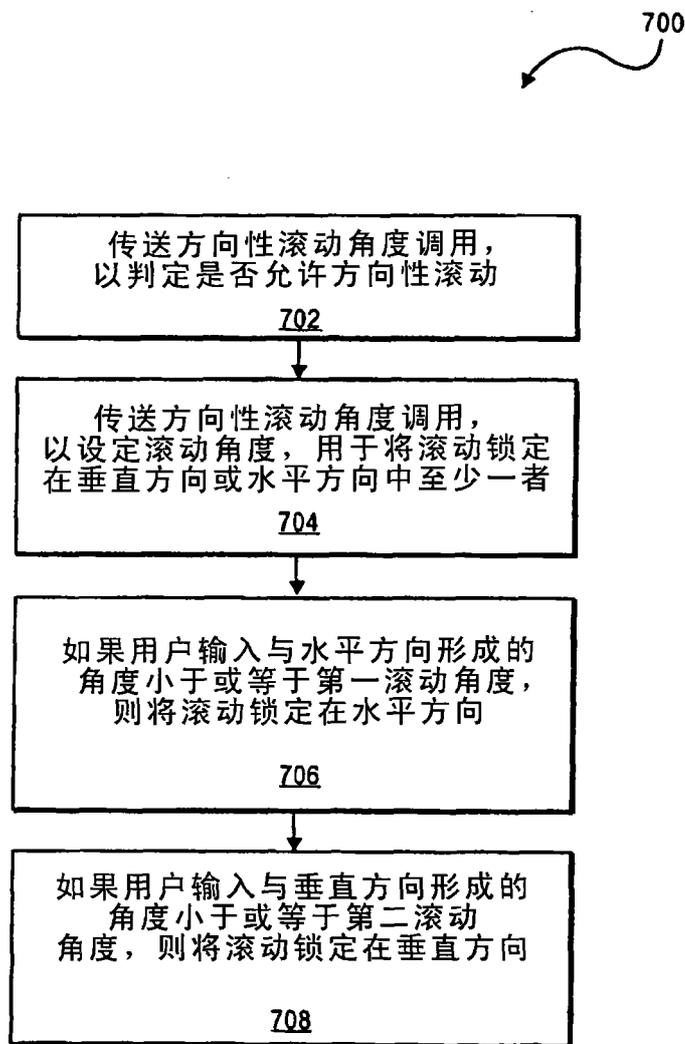


图7

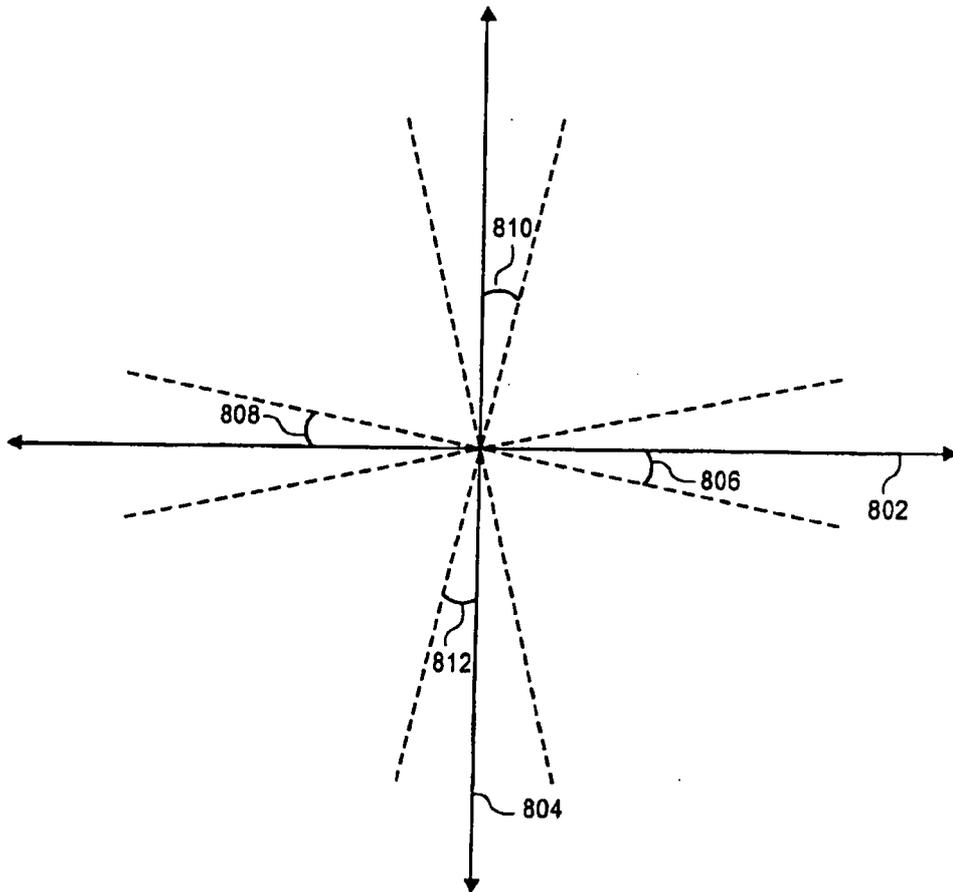


图8

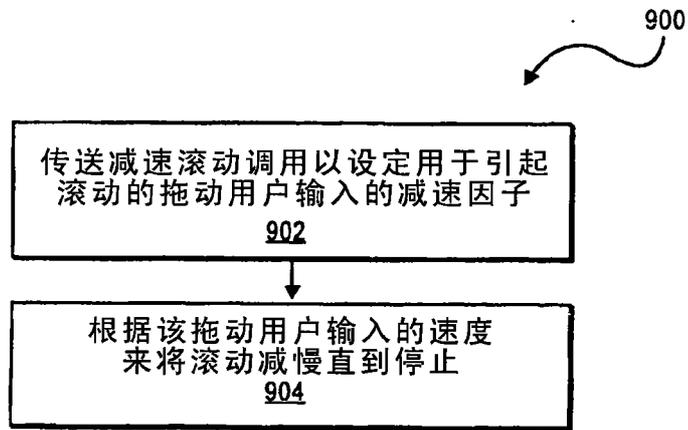


图9

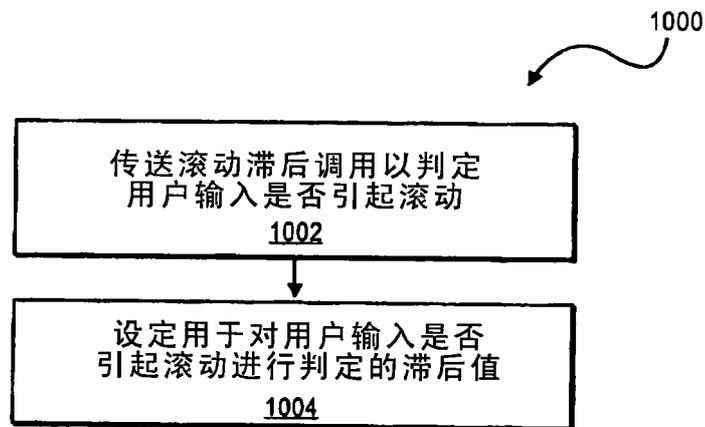


图10

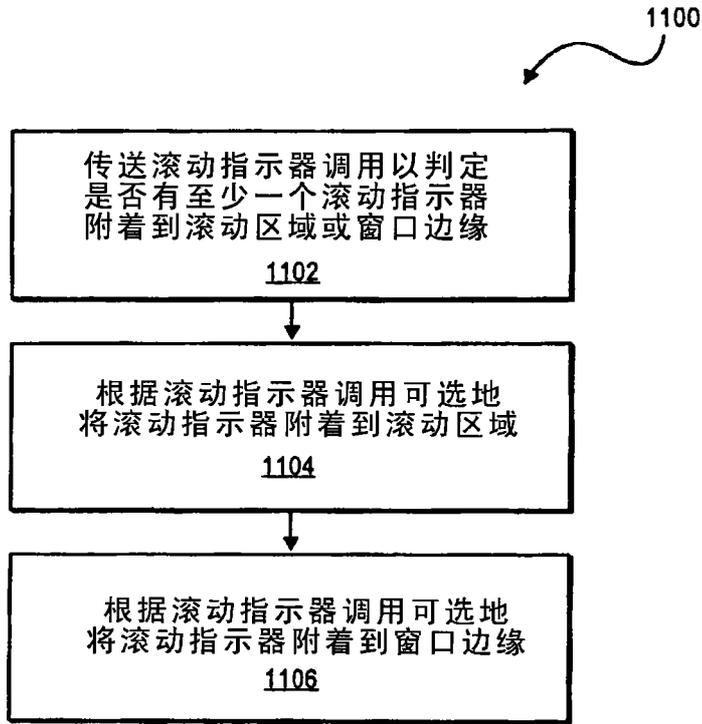


图11

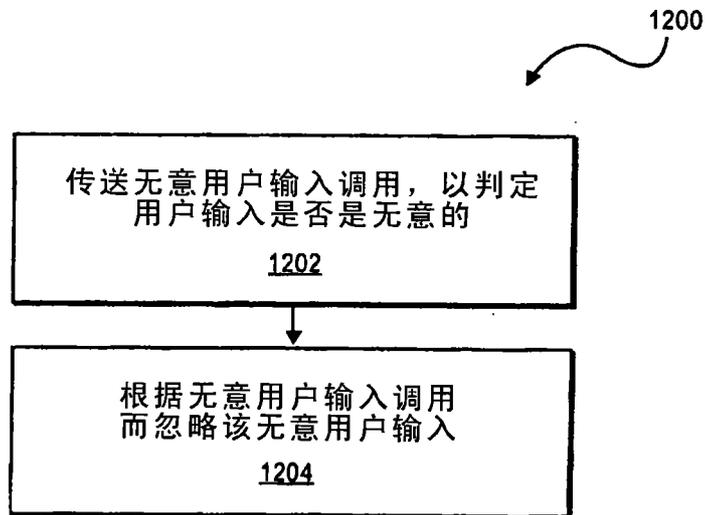


图12

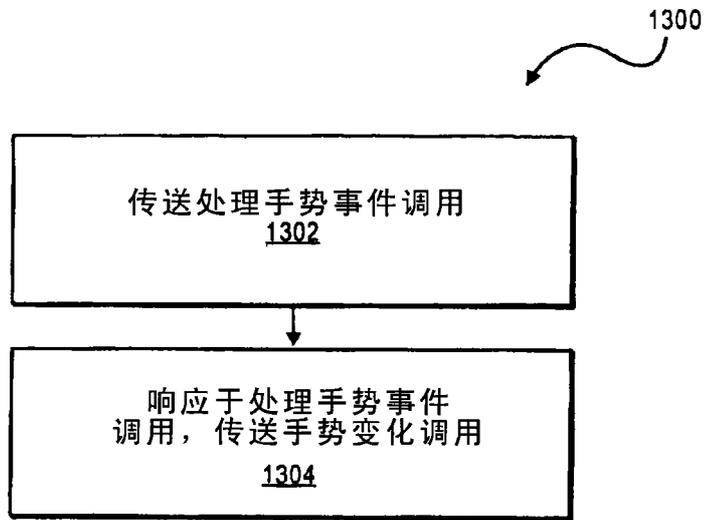


图13

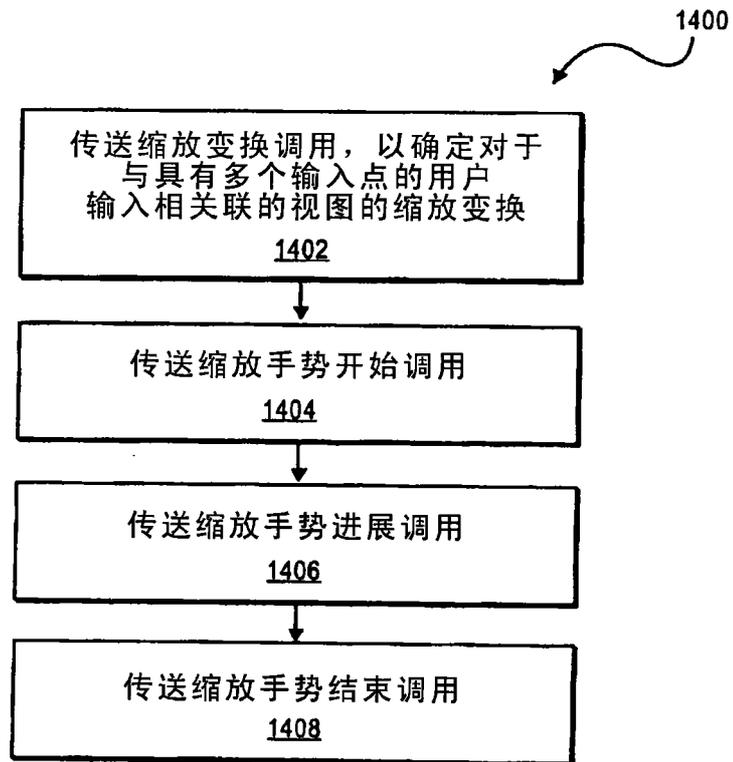


图14

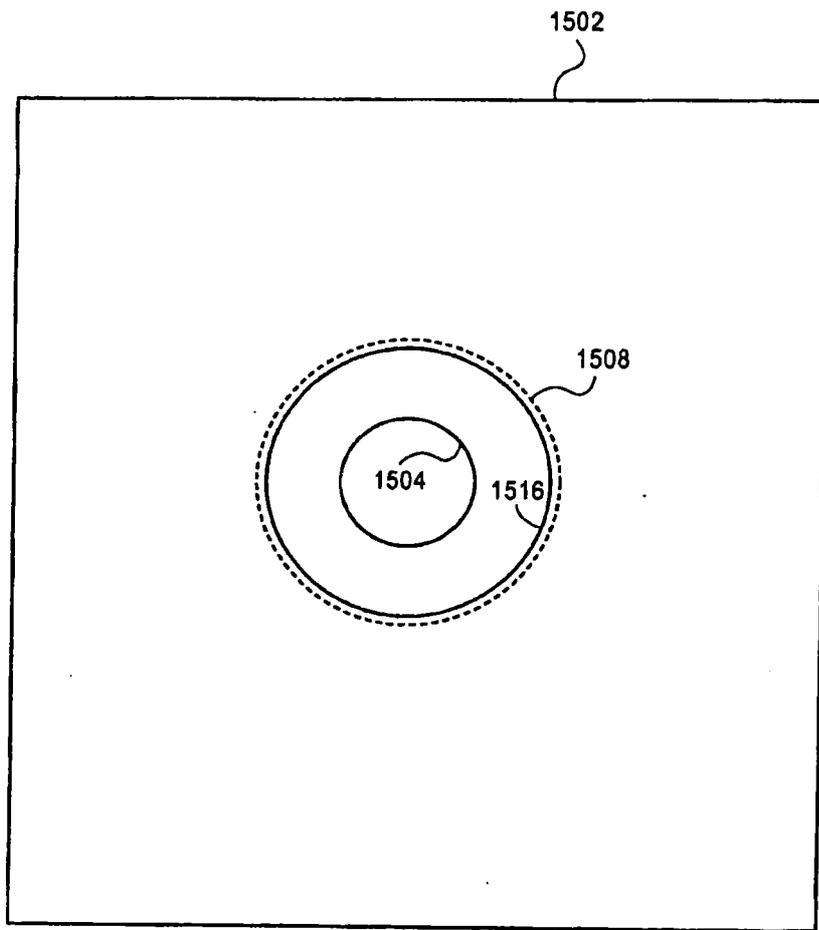


图15

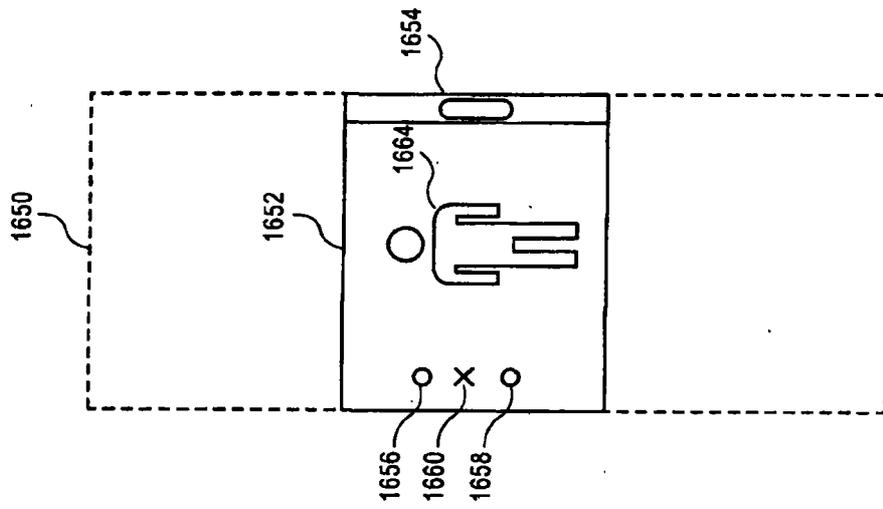


图16B

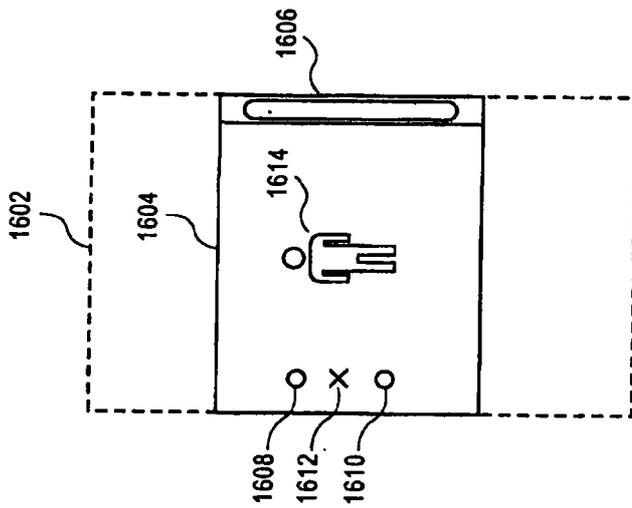


图16A

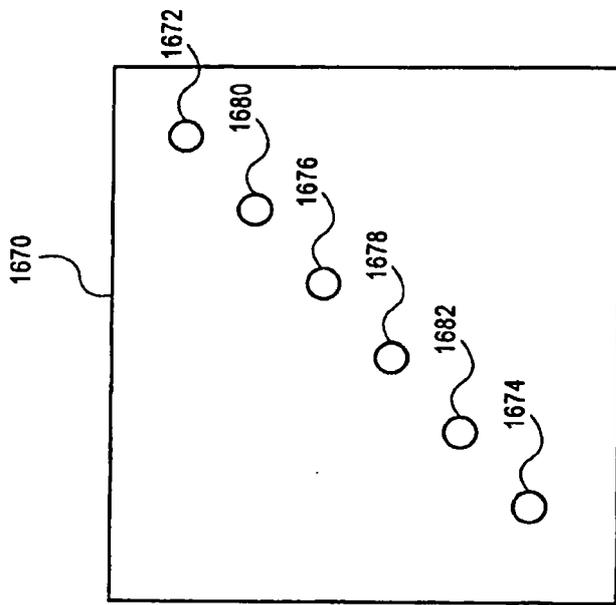


图160

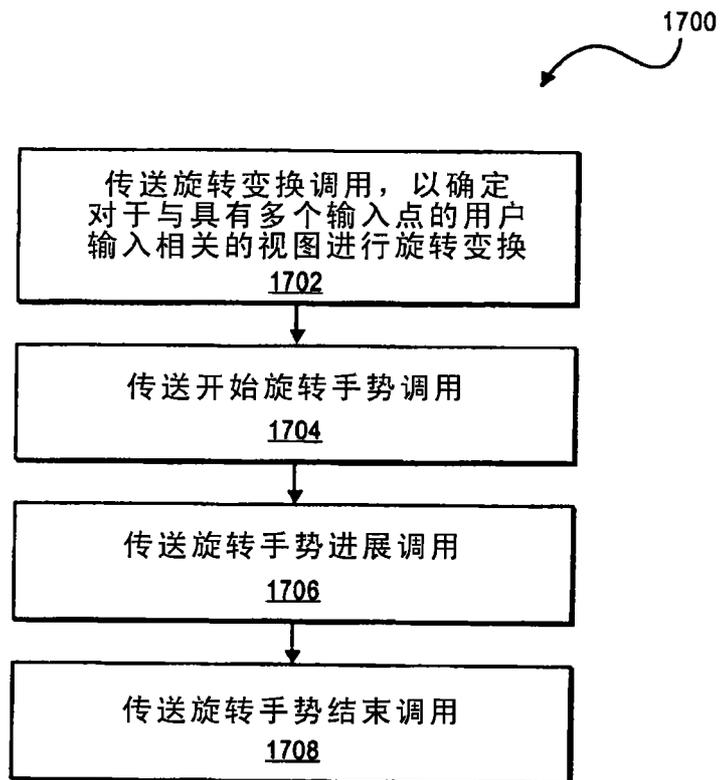


图17

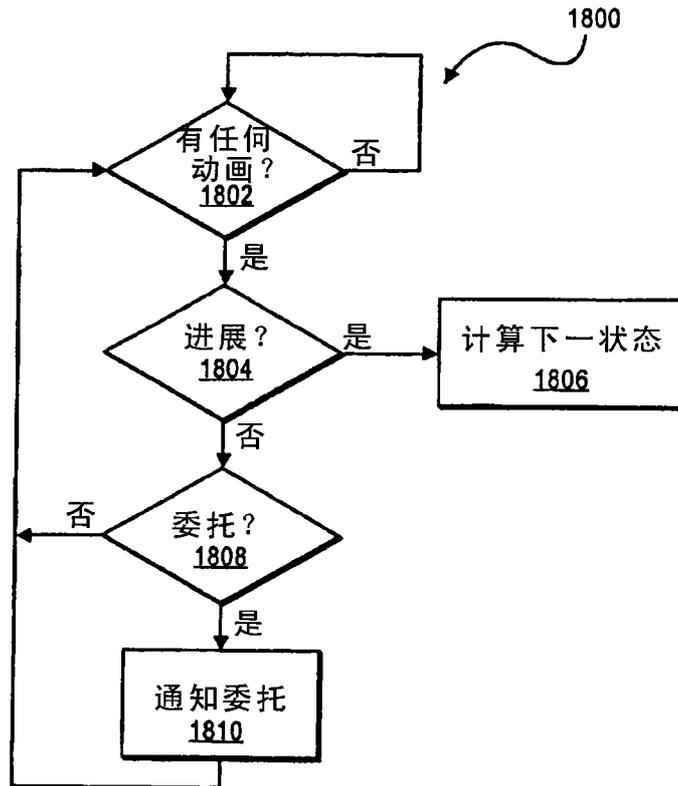


图18

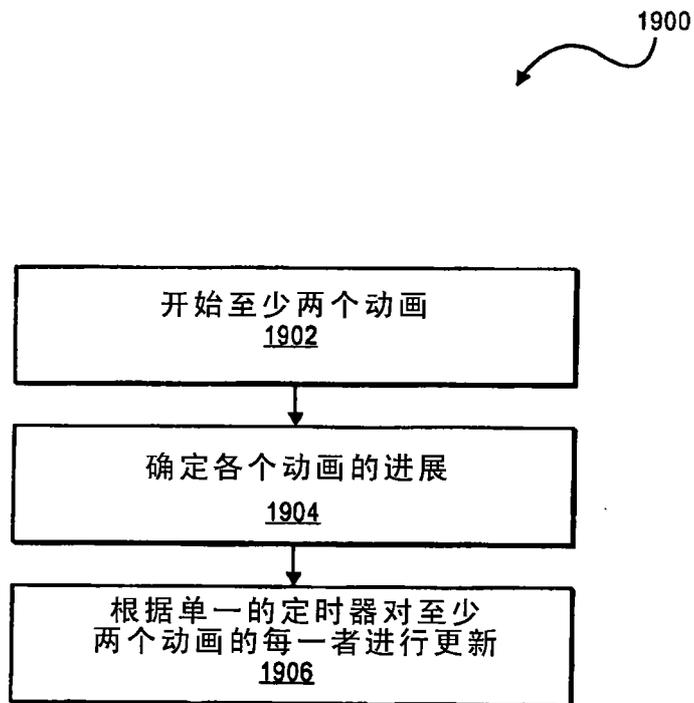


图19

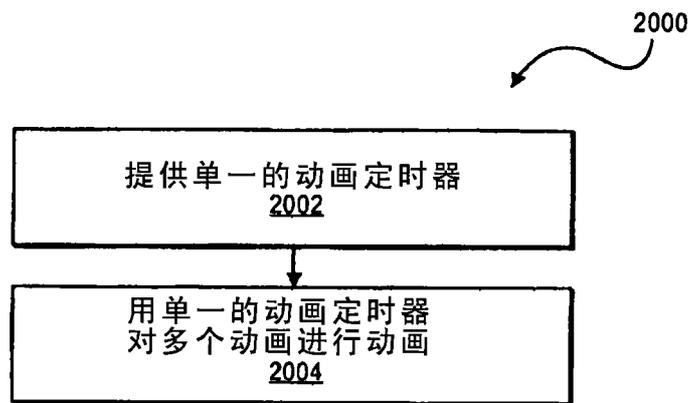


图20

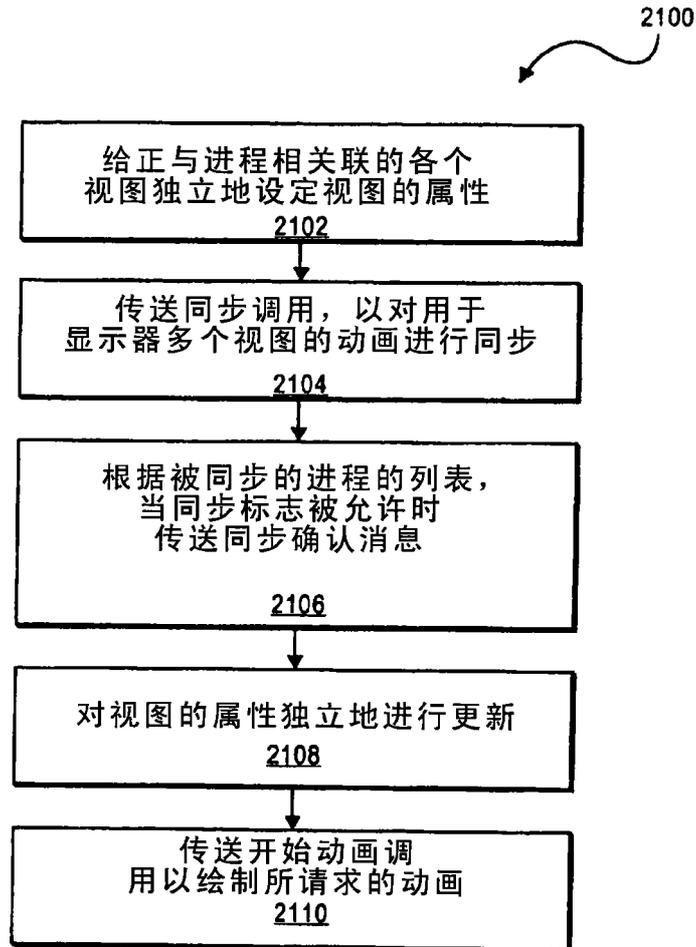


图21

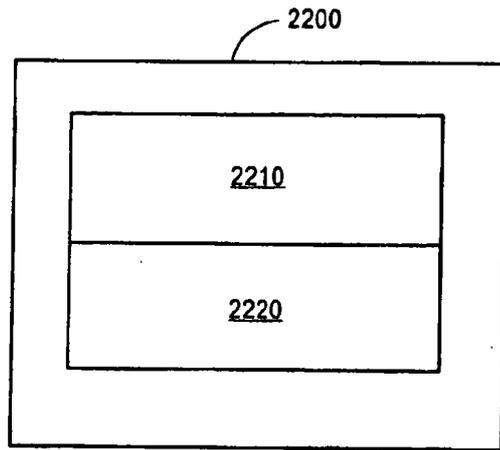


图22A

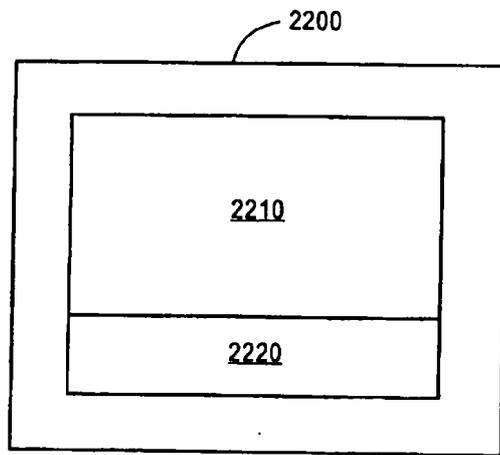


图22B

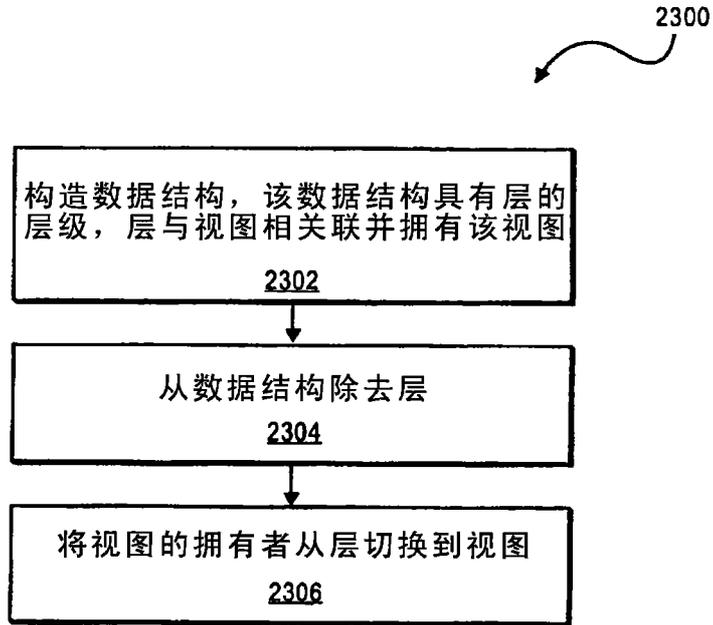


图23

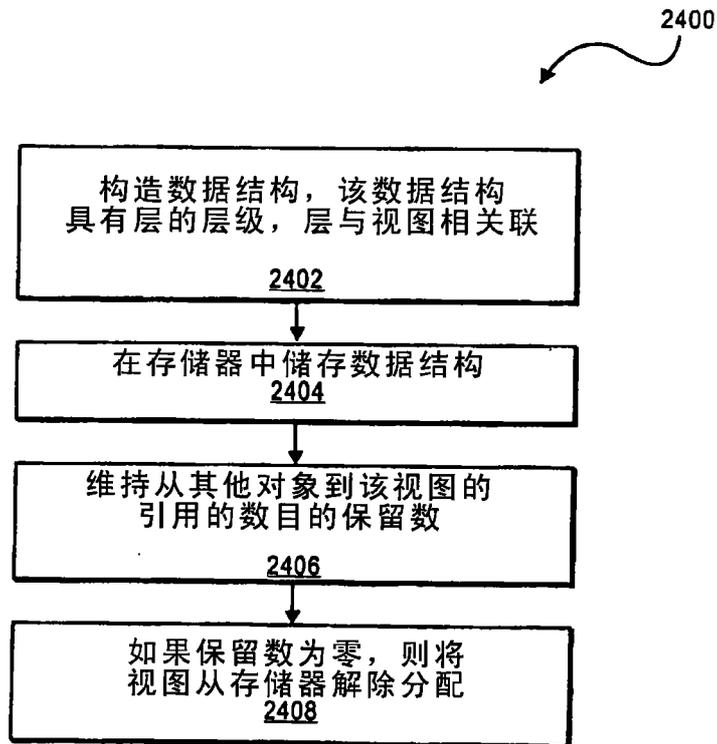


图24

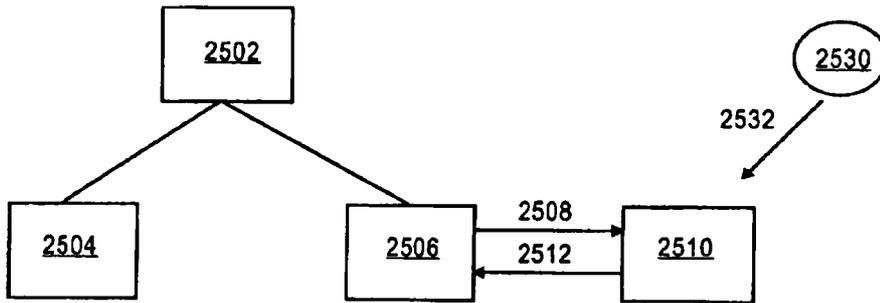


图25A

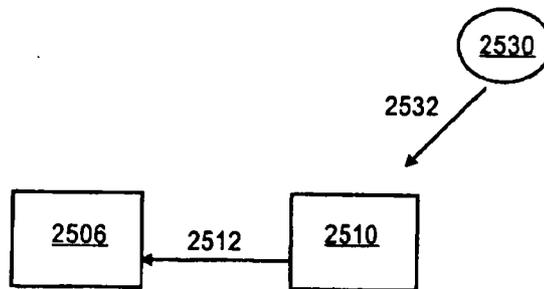


图25B

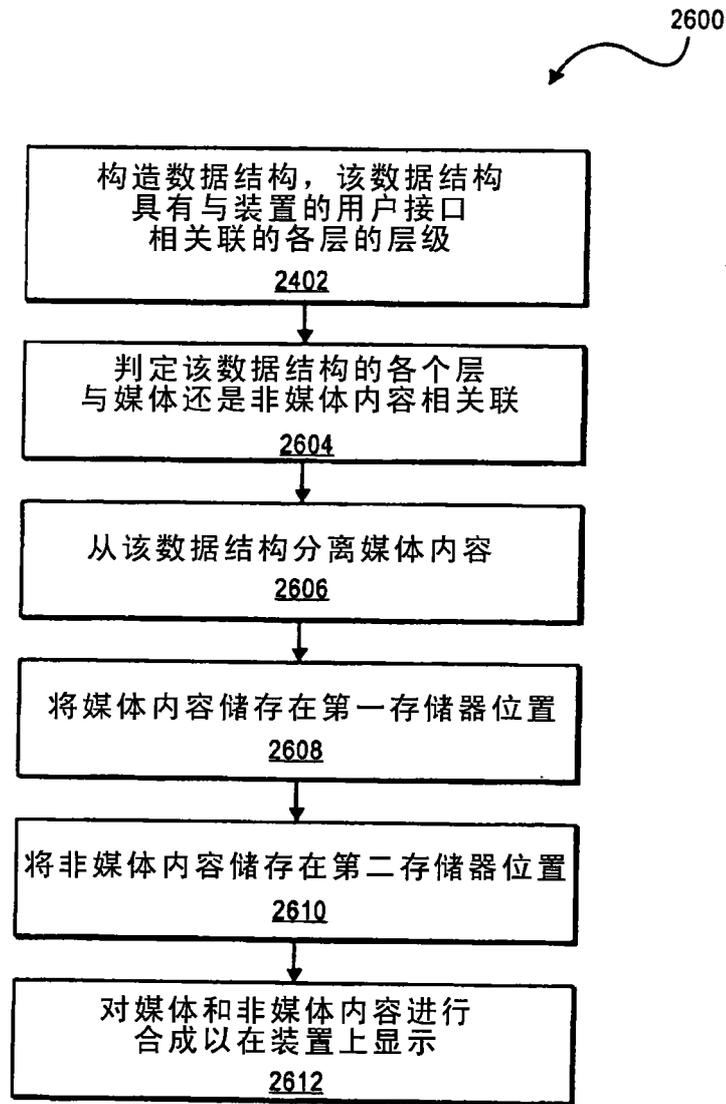


图26

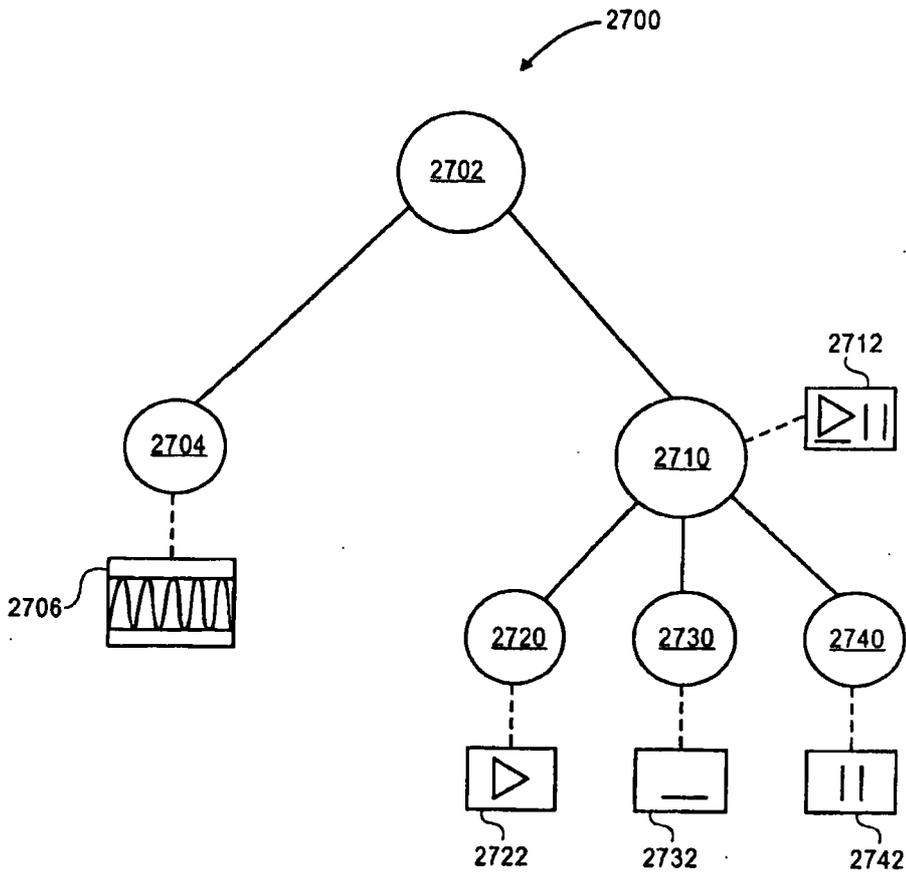


图27

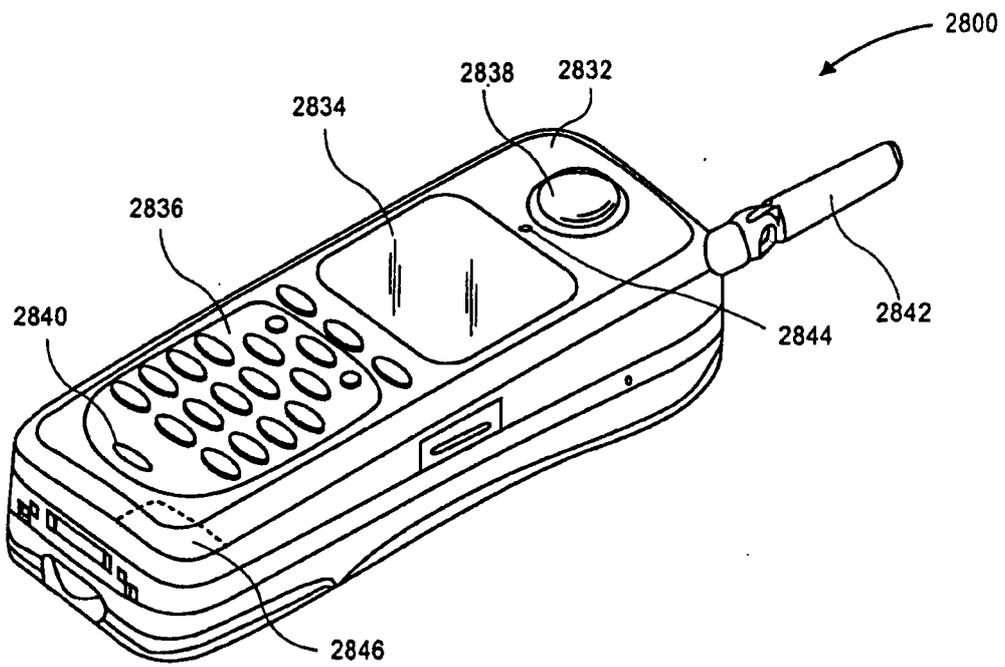


图28

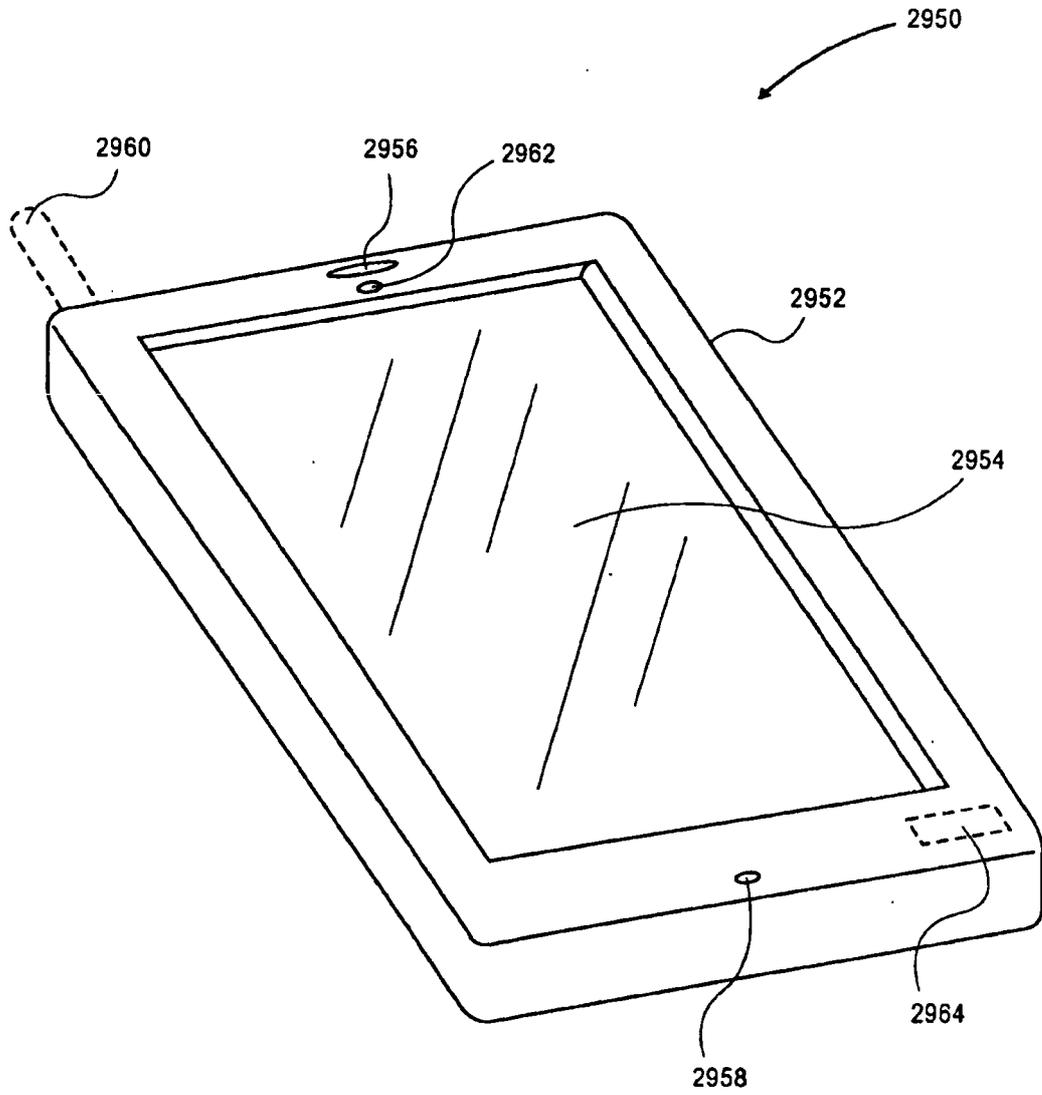


图29

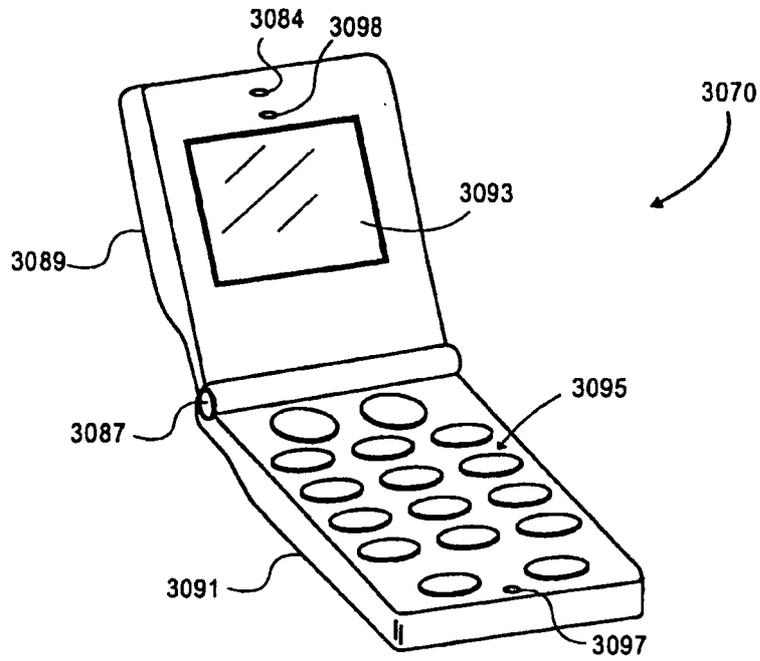


图30A

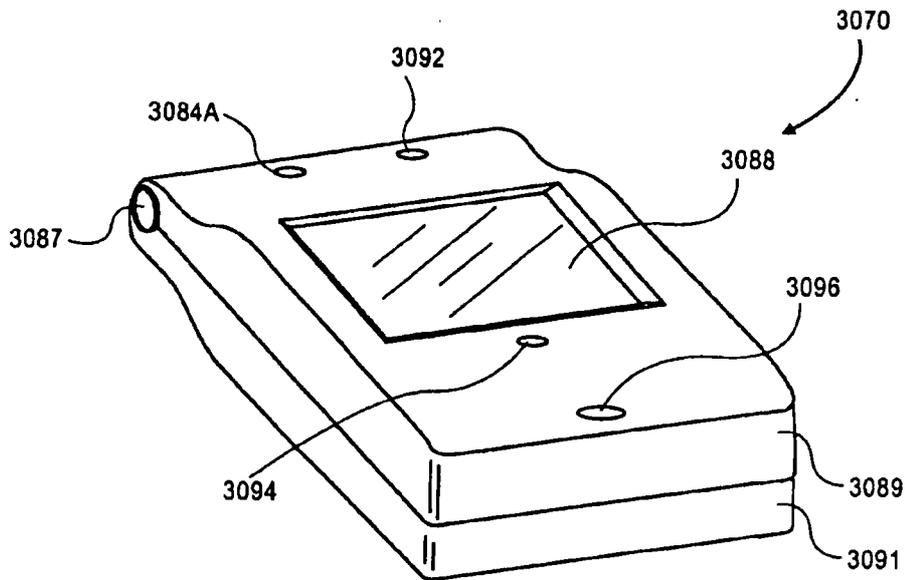


图30B

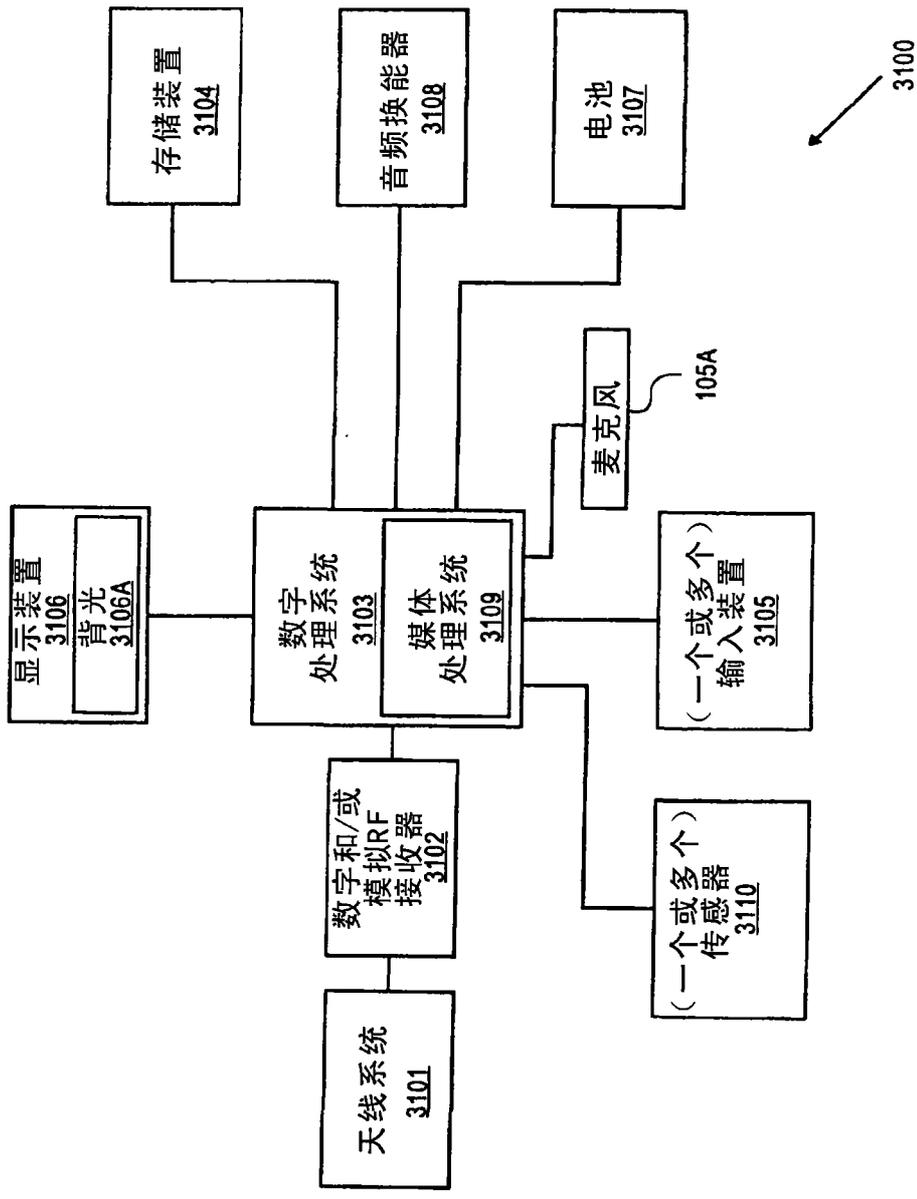


图31

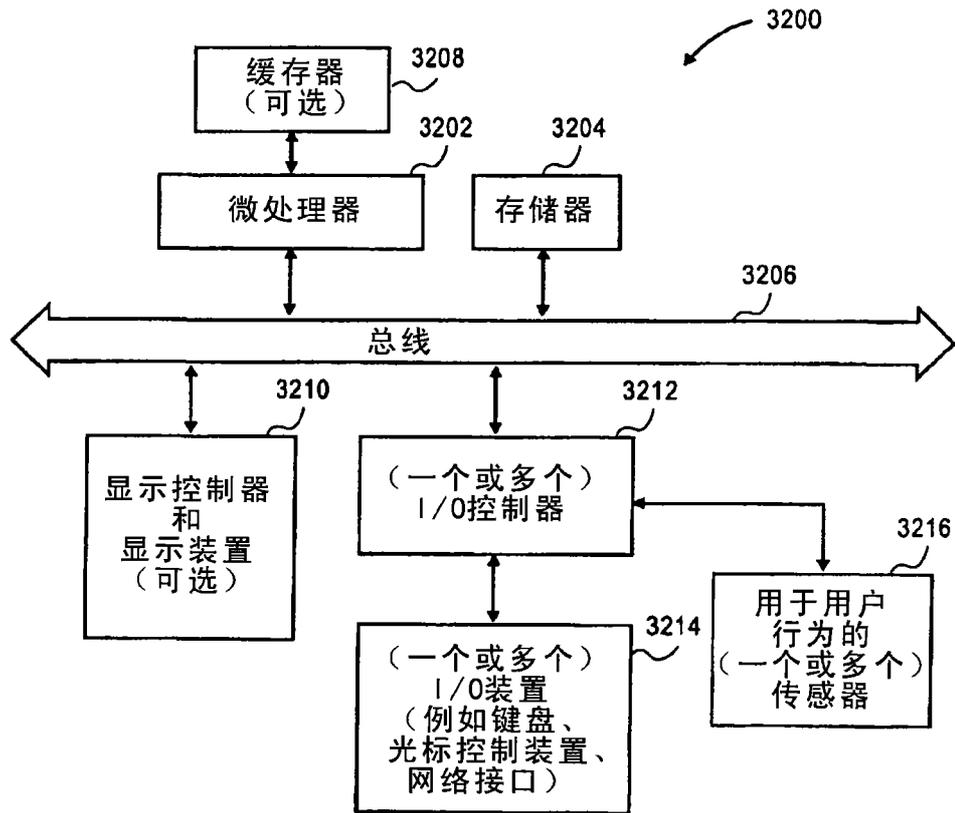


图32

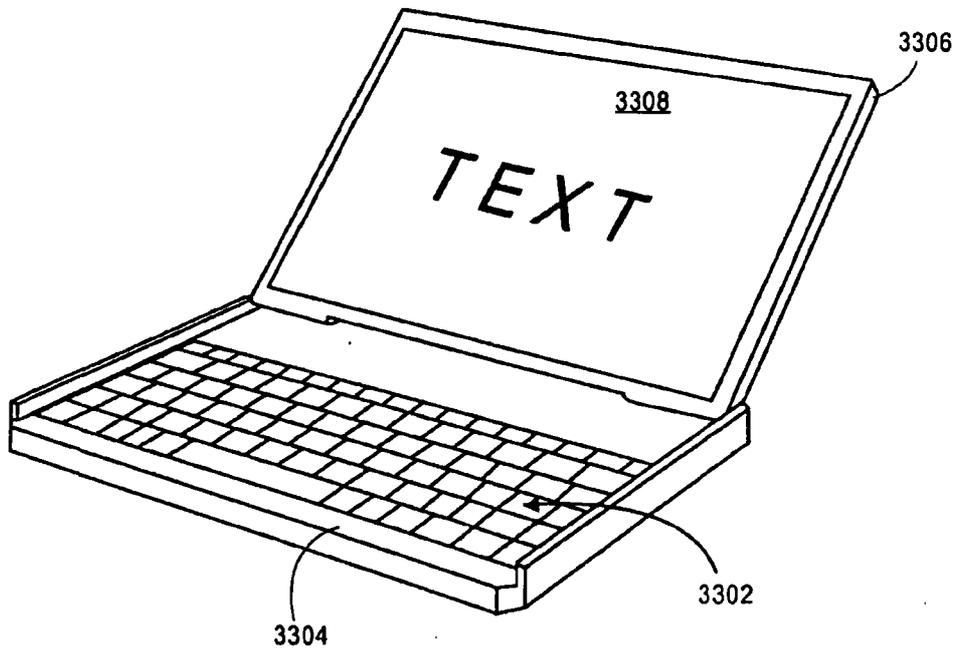


图33A

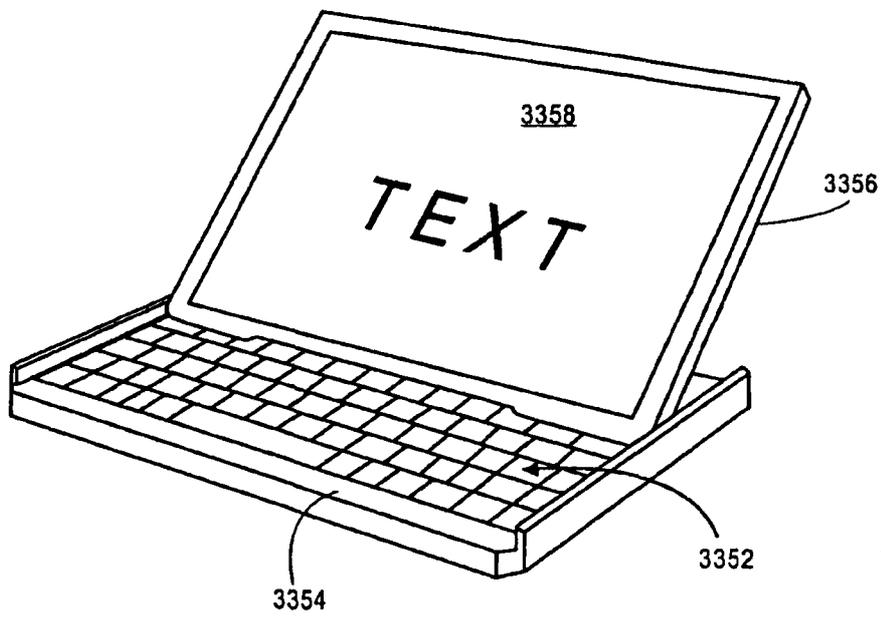


图33B

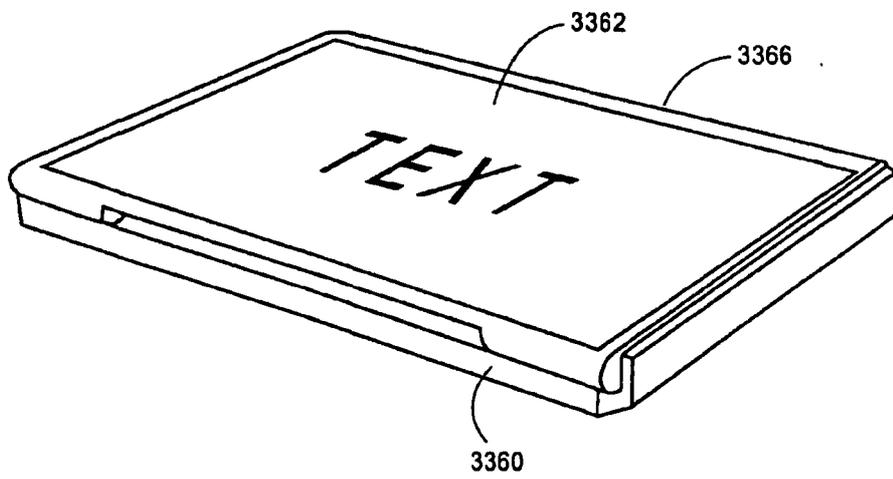


图33C

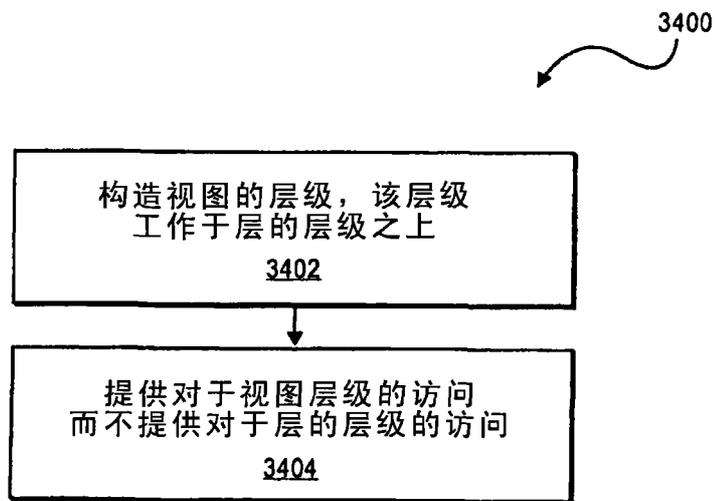


图34