



(12) 发明专利

(10) 授权公告号 CN 102804186 B

(45) 授权公告日 2015. 08. 19

(21) 申请号 201080027323. X

G06F 9/44(2006. 01)

(22) 申请日 2010. 06. 18

(56) 对比文件

(30) 优先权数据

12/488, 201 2009. 06. 19 US

W0 01/95103 A3, 2003. 09. 04,

US 2009/0013259 A1, 2009. 01. 08,

US 2005/0131657 A1, 2005. 06. 16,

CN 101198957 A, 2008. 06. 11,

Qfinsoft Technologies inc., QFIN 4.0 用户

手册. 《QFIN 4.0 用户手册》. 2009,

(85) PCT国际申请进入国家阶段日

2011. 12. 19

(86) PCT国际申请的申请数据

PCT/US2010/039268 2010. 06. 18

审查员 欧晓丹

(87) PCT国际申请的公布数据

W02010/148364 EN 2010. 12. 23

(73) 专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72) 发明人 B·C·贝克曼 D·G·格林

V·米塔尔 D·E·鲁宾

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 黄嵩泉

(51) Int. Cl.

G06F 17/50(2006. 01)

G06F 17/40(2006. 01)

G06F 3/14(2006. 01)

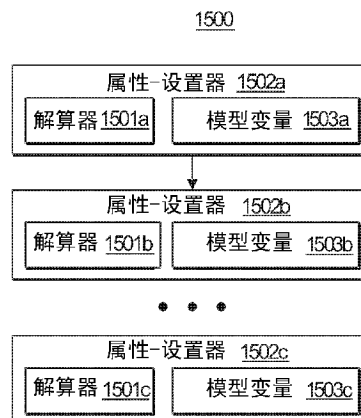
权利要求书2页 说明书40页 附图38页

(54) 发明名称

基于解算器的可视化框架

(57) 摘要

可视化框架可包括解算器。可以使用解算器来确定视图合成的视图组件的属性。在某些情况下,可以使用诸如依赖树之类的关系结构来显式地编写解算器。在某些情况下,可以基于具有解算器的属性-设置器调用其他具有解算器的属性-设置器来隐式地编写解算器。



1. 一种用于使用定义视图合成的一个或多个视图组件的属性的多个模型变量的计算机实现的方法,所述方法包括:

使用所述多个模型变量中的已知的第一模型变量(1503a)来求解(1401)所述多个模型变量中的未知的第二模型变量(1503b)的动作,所述第二模型变量(1503b)定义所述视图合成的第一视图组件的第一属性,所述已知的第一模型变量定义所述视图合成的第二视图组件的第一属性,其中,由属性-设置器为所述第一视图组件的所述第一属性调用的解算器执行所述使用所述多个模型变量中的所述已知的第一模型变量来求解所述多个模型变量中的所述未知的第二模型变量的动作,所述第一视图组件是所述第二视图组件的子;

将所述视图合成的所述第一视图组件的所述第一属性设置(1403)为所求解的第二模型变量(1503b)的值的动作,其中,所述第一视图组件的所述第一属性的所述属性-设置器执行将所述视图合成的所述第一视图组件的所述第一属性设置为所求解的第二模型变量的值的动作,所述第一视图组件的所述第一属性的所述属性-设置器被所述第二视图组件的所述第一属性的属性-设置器调用;以及

呈现(1404)包括所述第一视图组件的所述视图合成的动作。

2. 根据权利要求1所述的计算机实现的方法,其特征在于,所述已知的第一模型变量定义所述第一视图组件的第二属性。

3. 根据权利要求2所述的计算机实现的方法,其特征在于,由属性-设置器为所述第一视图组件的所述第一属性调用的解算器执行所述使用所述多个模型变量中的所述已知的第一模型变量来求解所述多个模型变量中的所述未知的第二模型变量的动作;以及,其中,所述第一视图组件的所述第一属性的所述属性-设置器执行将所述视图合成的所述第一视图组件的所述第一属性设置为所求解的第二模型变量的值的动作。

4. 根据权利要求3所述的计算机实现的方法,其特征在于,所述第一视图组件的所述第一属性的所述属性-设置器被所述第一视图组件的所述第二属性的属性-设置器调用。

5. 根据权利要求1所述的计算机实现的方法,其特征在于,由属性-设置器为所述第一视图组件的所述第一属性调用的解算器执行所述使用所述多个模型变量中的所述已知的第一模型变量来求解所述多个模型变量中的所述未知的第二模型变量的动作;以及,其中,所述第一视图组件的所述第一属性的所述属性-设置器执行将所述视图合成的所述第一视图组件的所述第一属性设置为所求解的第二模型变量的值的动作。

6. 一种用于使用定义视图合成的一个或多个视图组件的属性的多个模型变量的计算机实现的方法,包括:

调用(1601)第一属性-设置器(1502a),所述第一属性-设置器(1502a)被配置成设置(1602)视图合成的至少一个视图组件的至少一个属性,并调用(1603)第二属性-设置器(1502b),其中,所述第一属性-设置器被配置成设置所述视图合成的第二视图组件的第一属性,所述第二属性-设置器(1502b)被配置成:

调用(1604)被配置成对定义所述视图合成的一个或多个视图组件的属性的多个模型变量的未知的第一模型变量进行求解的解算器(1501b);以及

将所述视图合成的第一视图组件的第一属性设置(1605)为所求解的第一模型变量的值,所述第一视图组件是所述第二视图组件的子。

7. 根据权利要求6所述的方法,其特征在于,所述第一属性-设置器被配置成设置所述

视图合成的所述第一视图组件的第二属性。

基于解算器的可视化框架

背景技术

[0001] 向人传达信息的最有效的方式常常是在视觉上进行的。因此,数百万的人处理各种各样的视觉项目,以便传达或接收信息,以及以便协作。这样的视觉项目可包括,例如,概念简图、工程制图、材料清单、描绘诸如建筑物或分子结构之类的各种结构的三维模型、培训材料、带插图的安装说明书、规划图等等。

[0002] 最近,使用例如计算机辅助设计(CAD)以及实体建模应用程序来以电子方式构建这些视觉项目。这些应用程序常常允许作者将数据和约束附加于几何形状中。例如,用于构建材料清单的应用程序可以允许将诸如部件号和供应商之类的属性与每一部件、两个组件之间的最大角度等等相关联。构建圆剧场的电子版本的应用程序可以具有用于指定座位之间的最小间隙的工具等等。

[0003] 这样的应用程序对于设计和技术的进步贡献巨大。然而,任何给定应用程序都具有与可以在视觉上传达的信息的类型、如何在视觉上传达该信息、或可以归因于各种视觉表示的数据和行为的范围有关的限制。如果要修改该应用程序以超出这些限制,则计算机程序员通常将要创作扩展该应用程序的能力的新应用程序,或提供全新的应用程序。同样,对于有多少用户(除模型的实际作者以外)可以操纵该模型来测试各种情形也有限制。

发明内容

[0004] 此处所描述的实施例涉及其中解算器可以被用来确定视图组件的属性的可视化框架。在某些情况下,可以使用诸如依赖树之类的关系结构来显式地编写解算器。在某些情况下,可以基于具有解算器的属性-设置器调用其他具有解算器的属性-设置器来隐式地编写解算器。这可以允许作者更快速地创建和修改视图合成。

[0005] 本概述并不旨在标识出所要求保护的主题的关键特征或必要特征,也不旨在用于帮助确定所要求保护的主题的范围。

[0006] 附图简述

[0007] 为了描述可以获得本发明的上文所列举的及其他优点和特征的方式,下面将通过参考附图来呈现对各实施例的更具体的描述。可以理解,这些附图只描绘了示例实施例,并且因此不被认为是对其范围的限制,将通过使用附图并利用附加特征和细节来描述和解释各实施例,在附图中:

[0008] 图1示出了其中可以使用本发明的原理的环境,包括构建取决于输入数据的视图合成的数据驱动的合成框架;

[0009] 图2示出了表示图1的环境的一个示例的管道环境;

[0010] 图3示意性地示出了图2的管道的数据部分的实施例;

[0011] 图4示意性地示出了图2的管道的分析法部分的实施例;

[0012] 图5示意性地示出了图2的管道的视图部分的实施例;

[0013] 图6示意性地示出了能够枚举数据流的元素的全部或子集的数据流对象;

[0014] 图7示出了可以通过图2的管道来构建的视图合成的呈现;

- [0015] 图 8 示出了用于使用图 2 的管道环境来生成视图合成的方法的流程图；
- [0016] 图 9 示出了用于使用图 2 的管道环境响应于与视图合成的用户交互来重新生成视图合成的方法的流程图；
- [0017] 图 10 进一步详细地示意性地示出了图 4 的分析法部分的解算器，包括专门解算器的集合；
- [0018] 图 11 示出了通过协调专门解算器集合的动作来对未知模型参数进行求解的图 10 的解算器的流程图；
- [0019] 图 12 示意性地示出了可以表示图 10 的解算器的示例的解算器环境；
- [0020] 图 13 示出了用于使用图 12 的解算器环境来对模型分析法进行求解的方法的流程图；
- [0021] 图 14 示出了用于使用图 10 的解算器环境来对模型变量进行求解的方法的流程图；
- [0022] 图 15 示意性地示了解算器环境的实施例；
- [0023] 图 16 示出了可以由如图 15 所示的解算器环境执行的方法的流程图；
- [0024] 图 17 示出了扩展了图 7 的示例的集成视图合成的呈现；
- [0025] 图 18A 示出了其中多个视觉项目被用表示相对应的视觉项目可以与执行滚动进行交互的视觉线索来装饰的视图合成；
- [0026] 图 18B 示出了在滚动交互之后图 18A 的视图合成；
- [0027] 图 19A 示出了其中允许放大交互性的视图合成；
- [0028] 图 19B 示出了在放大交互性之后图 19A 的视图合成；
- [0029] 图 20 以美国地图的形式示出了视图合成，其中，每一个州的高度表示数据，包括视觉线索的一些州视觉项目指示与该州视觉项目的可能交互性；
- [0030] 图 21 以组合的相互关联的饼图和柱状图的形式示出了视图合成，包括示出了可能的交互性的视觉线索；
- [0031] 图 22 以飓风路线前进图的形式示出了视图合成，其中各种视觉线索标识交互性；
- [0032] 图 23 示出了用于在视图合成中提供交互性的方法的流程图；
- [0033] 图 24 示出了其中各种视觉项目可以被集成和合并的用户界面；
- [0034] 图 25 示出了集成的第一阶段，其中，仅螺旋被示出为数据将被映射到其上的形状；
- [0035] 图 26 示出了集成的第二阶段，其中，图 25 的螺旋被绑定到一数据系列；
- [0036] 图 27 示出了集成的第三阶段，其中，图 25 的螺旋被绑定到两个数据系列；
- [0037] 图 28 示出了集成的最后阶段，其中，图 25 的螺旋被绑定到所示出的图表。
- [0038] 图 29 示出了架子布局的可视化，并只表示此处所描述的原理可以适用的无数应用程序中的一个；
- [0039] 图 30 示出了此处所描述的原理也可以适用的市区规划的可视化；
- [0040] 图 31 示出了本发明的原理可以适用的从而创建更动态的学习环境的比较了儿童的教育常规可视化；
- [0041] 图 32 示出了本发明的原理可以适用的从而创建更动态的学习环境的比较了人口密度的常规可视化；

- [0042] 图 33 示出了应用于第一参数目标的视图组件的可视化；
- [0043] 图 34 示出了应用于第二参数目标的如图 33 所示的视图组件的可视化；以及
- [0044] 图 35 示出了图 2 的分类组件可以在其中操作的分类环境；
- [0045] 图 36 示出了图 35 的成员项目的分类的示例；
- [0046] 图 37A 到 37C 示出了相关的类别的分类的三个示例；
- [0047] 图 38 示出了包括多个属性的成员项目；
- [0048] 图 39 示出了领域特定的分类并表示图 35 的领域指定的分类的一个示例；
- [0049] 图 40 示出了用于导航和使用分析法的方法的流程图；
- [0050] 图 41 示出了用于使用分析法来进行搜索的方法的流程图；以及
- [0051] 图 42 示出了表示其中可以实现图 1 的合成框架（或其各部分）的环境的计算系统。
- [0052] 详细描述

[0053] 图 1 示出了使用数据驱动的分析法和分析结果的可视化的视觉合成环境 100。环境 100（下面也叫做“管道”）包括合成框架 110，该合成框架 110 执行独立于视图构造 130 的问题领域而执行的逻辑。例如，同一个合成框架 110 可以用来合成用于城市规划、分子模型、杂货店货架布局、机器性能或装配分析或其他领域特定的呈现的交互式视图合成。如下文所描述的，可以使用分析法来搜索并在各种方案中探索。然而，首先，将详细描述基本合成框架 110。

[0054] 合成框架 110 使用以领域特定的方式分类学地组织的领域特定的数据 120 来执行分析法 121，以构建领域特定的实际视图构造 130（此处也叫做“视图合成”）。因此，通过改变域专用数据 120，可以使用同一个合成框架 110 来构建任意数量的不同领域的视图合成，而非必须重新编码合成框架 110 本身。如此，通过改变数据而非重新编码和重新编译，管道 100 的合成框架 110 可以潜在地适用于无限多个问题领域，或至少适用于各种各样的问题领域。然后，可以将视图合成 130 作为指令提供给适当的 2-D 或 3-D 呈现模块。此处所描述的体系结构还允许将预先存在的视图合成作为构件方便地合并到新视图合成中。在一个实施例中，可以将多个视图合成包括在集成视图合成中，以方便地在对模型的两个可能的解决方案之间进行比较。

[0055] 图 2 示出了以管道环境 200 的形式存在的合成框架 110 的示例体系结构。管道环境 200 包括管道 201 本身，以及其他。管道 201 包括数据部分 210、分析法部分 220，以及视图部分 230，分别参考随后的图 3 到 5 以及伴随的描述来详细地对它们进行了描述。现在，在一般水平，管道 201 的数据部分 210 可以接受各种不同类型的数据，以规范形式将该数据提供给管道 201 的分析法部分 220。分析法部分 220 将该数据绑定到各种模型参数，并使用模型分析法求解模型参数中的未知数。然后，将各种参数值提供给视图部分 230，该视图部分 230 使用模型参数的这些值来构建合成视图。管道环境 200 还包括创作组件 240，该创作组件 240 允许管道 201 的作者或其他用户制定和 / 或选择向管道 201 提供的数据。例如，创作组件 240 可以用来向数据部分 210（由输入数据 211 表示）、分析法部分 220（由分析法数据 221 表示）以及视图部分 230（由视图数据 231 表示）中的每一个部分提供数据。各种数据 211, 221 和 231 表示图 1 的领域特定的数据 120 的示例，下面将更详细地对它们进行描述。创作组件 240 支持提供各种数据，所述各种数据包括例如数据模式，供模型使用

的实际数据,从外部源带进的数据的位置或可能位置的范围,视觉(图形或动画)对象,可以对画面、建模语句(例如,视图、方程、约束)、绑定等等执行的用户界面交互。

[0056] 在一个实施例中,创作组件只是由总体管理器组件(在图2中未示出,但是由图1的合成框架110表示)所提供的功能的一个部分。该管理器是响应于事件(诸如用户交互事件、外部数据事件,以及来自诸如解算器、操作系统等等之类的其他组件中的任何一个组件的事件)来控制 and 排序所有其他组件(诸如数据连接器、解算器、查看器等等)的操作的总体指挥器。

[0057] 创作组件240还包括允许执行搜索的搜索工具242。搜索工具242能够利用管道201的数据驱动的分析能力,以便执行复杂的搜索操作。例如,在某些情况下,搜索的一个或多个参数可能首先需要被求解,以便完成搜索。

[0058] 作为示例,假设数据驱动的分析法是城市图,并且一些分析法能够对城市中特定坐标处的典型的噪音级别进行求解。在该情况下,一个正在搜索房地产的人可以不仅仅对平方尺、价格范围、房间数量等等典型的搜索参数执行搜索,而且还可对分析强度参数进行搜索。例如,尽管有无限的可以应用原理的方式,但是,现在将提供如何可以实现对房地产的灵活搜索的选定的几个不同的示例。

[0059] 作为指定搜索参数的一部分,用户可以指示需要下面或其他搜索项目中的任何一个或多个:

[0060] 1) 只有那些体验到低于某一上限的平均噪音级别的房屋;

[0061] 2) 只有那些在星期一到星期四中每一天到用户的工作单位和体育馆具有累积300分钟通勤时间或更短的房屋,

[0062] 3) 只有那些在五分之一英里内在任何道路上体验到小于某一阈值交通流量并且被预测在未来10年保持该交通流量级别之下的房屋,

[0063] 4) 只有那些在一年中的上午9:15之后的任何时间都不在高山的阴影里的房屋,

[0064] 5) 只有那些在地产处具有足够的原有树木以便在10年时间内,树木将遮蔽屋顶的至少50%的面积 of 的房屋,

[0065] 6) 等等。

[0066] 这样的房地产搜索不能通过使用常规技术来轻松地实现。在这些示例中的每一个中,被请求的房屋搜索数据可能不存在,但是,此处所描述的原理可以允许当搜索正在被执行时在运行时生成各种自定义参数的搜索数据。同样,搜索还可以利用预先被求解的任何搜索数据。这允许为用户提供各种搜索和探查能力,并打开探索要被解决的问题的全新的方式。现在将描述更多关于支持这些类型的搜索的基础数据驱动的分析法机制。

[0067] 传统上,交互式视图合成应用程序的生命周期涉及两个关键时间:创作时间和使用时间。在创作时,交互式视图合成应用程序的功能由程序员编码以提供所需领域特定的交互式视图合成。例如,室内设计应用程序的作者(例如,通常是计算机程序员)可能对允许用户执行室内设计专用的有限动作集的应用程序编写代码。

[0068] 在使用时,用户(例如,或许是业主或专业室内设计师)可以使用该应用程序来执行被硬编码到该应用程序中的一组有限动作中的任何一个或多个动作。在室内设计应用程序示例中,用户可以指定被显示的虚拟房间的尺寸,向房间添加家具及其他室内设计组件,或许旋转视图以获得房间的各种角度,设置每一项目的颜色,等等。然而,除非用户是不介

意对室内设计应用程序进行反向工程和修改的程序员,否则用户仅限于被应用程序作者允许的有限动作集。例如,除非由应用程序提供,用户将不能使用应用程序自动地弄清楚哪一种窗户布局会使环境噪声最小化,根据“风水”规则房间布局表现得如何,或最小化太阳能贡献。

[0069] 然而,在图 2 的管道环境 200 中,创作组件 240 被用来向现有管道 201 提供数据,在现有管道 201 中,正是该数据驱动从定义输入数据,到定义分析模型,到定义在视图合成中如何可视化分析法的结果的整个过程。因此,不必执行任何代码编写就能使管道 201 适应各种领域和问题中的任何一个。只有提供给管道 201 的数据才是需要改变的,以便应用管道 201 以从不同问题领域可视化不同视图合成,或者或许调整对于现有领域的问题求解。管道环境 200 还可包括分类组件 260,该分类组件 260 组织、分类所提供的数据,并将其与管道 201 相关联。分类组件 260 可以对领域敏感。如此,室内设计领域、公路设计领域、建筑学领域、“风水”领域中每一个都可以具有可以被用来在数据中导航的不同的分类。这可以是有用的,因为,如下文所描述的,由于其中对管道环境 200 可用的数据集可能不断增加的合成活动,可以有大量的可用数据供筛选。

[0070] 此外,由于数据可以在使用时(即,运行时)以及在创作时改变,所以可以在运行时修改和/或扩展模型。如此,在创作模型和运行该模型之间有较少的(如果有的话)区别。由于所有创作都涉及编辑数据项,并且由于软件从数据运行其所有的行为,所以对数据的每个改变都立即影响行为,无需重新编码和重新编译。

[0071] 管道环境 200 还包括用户交互响应模块 250,该模块检测用户何时与显示的视图合成进行交互,并随后确定作为响应执行什么操作。例如,某些类型的交互可能不需要提供给管道 201 的数据的变化,并且因此不需要对视图合成进行更改。其他类型的交互可能改变数据 211,221,或 231 中的一个或多个。在该情况下,此新的或经过修改的数据可能导致新输入数据被提供给数据部分 210,可能需要分析法部分 220 对该输入数据进行重新分析,和/或可能需要视图部分 230 对视图合成进行重新可视化。

[0072] 因此,管道 201 可以用来将数据驱动的分析可视化扩展到或许无限多个问题领域,或至少各种各样的问题领域。此外,不必是程序员即可改变视图合成以解决各种问题。现在将参考图 3 的数据部分 300,图 4 的分析法部分 400,以及图 5 的视图部分 500 来按该顺序描述管道 201 的数据部分 210,分析法部分 220 和视图部分 230 中的每一部分。另外,对数据的分类是领域特定的,如此,允许对数据的组织对于领域中操作的那些数据更直观。从图 3 到 5 显而易见地看出,管道 201 可以构建成一系列转换组件,其中这些组件各自 1) 接收某些适当的输入数据,2) 响应于该输入数据,执行某一动作(诸如对输入数据执行转换),以及 3) 输出随后充当下一转换组件的输入数据的数据。

[0073] 管道 201 可以在客户机上、在服务器上实现,或甚至可以分布在客户机和服务器之间,这没有限制。例如,管道 201 可以在服务器上实现,并提供呈现指令作为输出。客户机侧的浏览器可以只根据从服务器接收到的呈现指令来进行呈现。在另一方面,管道 201 可以被包含在客户机上,创作和/或使用在客户机上执行。即使管道 201 完全在客户机上,管道 201 仍可以搜索客户机外部的数据源以查找适当的信息(例如,模型、连接器、规范化器、模式,及其他)。还有提供这两种方法的混合的实施例。例如,在一种这样的混合方法中,模型主存在服务器上,但是,web 浏览器模块被动态地在客户机上加载,以便使一些模型

的交互和查看逻辑在客户机上运行（如此，允许更丰富的并且更快的交互和视图）。

[0074] 图 3 只示出了图 2 的管道 201 的数据部分 300 的许多可能的实施例中的一种。数据部分 300 的各功能中的一个功能是以规范格式提供数据，该规范格式符合参考图 4 所讨论的管道的分析法部分 400 所理解的模式。数据部分包括访问异构数据 301 的数据访问组件 310。输入数据 301 在数据能（但是不必）以规范形式呈现给数据访问组件 310 的意义上可以是“异构的”。事实上，构建数据部分 300 以使得异构数据可以是各种格式。可以由模型访问和操作的不同类型的领域数据的示例包括文本和 XML 文档、表、列表、分层结构（树）、SQL 数据库查询结果、BI（商务智能）立方查询结果、各种格式的诸如 2D 图形和 3D 视觉模型之类的图形信息、以及其组合（即，合成）。此外，可以通过提供要被访问的数据的定义（例如，模式）来声明性地扩展可以被访问的数据的种类。因此，数据部分 300 允许到模型的各种异构输入，还支持可访问的数据类型的运行时、声明性扩展。

[0075] 在一个实施例中，数据访问部分 300 包括用于从许多不同的数据源获取数据的许多连接器。由于连接器的主要功能中的一个是对应的数据置于规范形式，这样的连接器在下文以及在附图中常常被称为“规范化器”。每一规范化器都可以理解其对应的数据源的专用应用程序编程接口（API）。规范化器还可以包括用于与该对应的 API 进行接口以从数据源读取和 / 或向数据源写入数据的对应逻辑。如此，规范化器在外部数据源和数据的存储器映像之间充当桥梁。

[0076] 数据访问组件 310 评估输入数据 301。如果输入数据已经是规范的，并且因此可以由分析法部分 400 处理，则输入数据可以作为规范数据 340 被直接提供，以便输入至分析法部分 400。

[0077] 然而，如果输入数据 301 不是规范的，则适当的数据规范化组件 330 能够将输入数据 301 转换为规范格式。数据规范化组件 330 实际上是数据规范化组件 330 的集合，每一个数据规范化组件 330 都能够将具有特定特征的输入数据转换为规范形式。规范化组件 330 的集合被示为包括四个规范化组件 331、332、333 和 334。然而，省略号 335 表示也可以有其他数量的规范化组件，或许甚至比所示出的四个更少。

[0078] 输入数据 301 甚至可以包括规范化器本身以及相关数据特征的标识。数据部分 300 然后可以注册相关数据特征，并向数据规范化组件集合 330 提供规范化组件，其中可以向可用的规范化组件添加规范化组件。如果稍后接收到具有那些相关特征的输入数据，则数据部分 310 可以将该输入数据分配给相关的规范化组件。也可以从外部源，诸如从 web 上的定义的组件库动态地查找规范化组件。例如，如果给定数据源的模式是已知的，但是所需的规范化器不存在，则可以从外部组件库定位规范化器，假设可以找到这样的库并且其中包含所需的组件。该管道也可以解析其模式仍未知的数据，并将解析结果与已知组件库中的模式信息进行比较，以尝试对该数据的类型进行动态判断，并且因此定位所需的规范化器组件。

[0079] 可另选地，并非输入数据包括所有规范化组件，输入数据可以改为提供定义了规范化转换的转换定义。集合 330 然后可以被配置成将该转换定义转换为对应的规范化组件，该规范化组件与零个或更多个标准默认规范化转换一起实施转换。这表示其中数据部分 300 消费输入数据并且不沿着管道向下提供对应的规范化的数据的情况的示例。然而，在或许大多数情况下，输入数据 301 导致生成对应的规范化的数据 340。

[0080] 在一个实施例中,数据访问组件 310 可以被配置成在输入数据的文件类型和 / 或格式类型的基础上将输入数据分配给数据规范化组件。其他特征可包括例如输入数据的源。可以将默认规范化组件分配给没有指定的对应的规范化组件的输入数据。默认规范化组件可以应用一组规则以尝试将输入数据规范化。如果默认规范化组件不能将该数据规范化,则该默认规范化组件可以触发图 2 的创作组件 240,提示用户提供输入数据的模式定义。如果模式定义还不存在,则创作组件 240 可以呈现模式定义助手,以帮助作者生成可以用来将输入数据转换为规范形式的对应的模式定义。一旦数据处于规范形式,则伴随该数据的模式提供对该数据的足够描述,使得管道 201 其余部分不需要新代码来解释该数据。相反,管道 201 包括能够根据可通过可访问的模式声明语言来表达的任何模式来解释数据的代码。

[0081] 一种类型的数据是数据流对象,如参考图 6 所示出的和描述的数据流对象 600。数据流对象 600 包括枚举模块 601,该枚举模块 601 能够枚举数据流 602 内的与数据流对象相关联的某一范围。该范围可以事实上包括数据流 602 的整个范围。另一方面,数据流 602 可以是“伪无限”或“部分地伪无限”数据流。在本描述和在权利要求书中,“伪无限”数据流是太大难以完全与正在处理数据流对象 600 的计算系统的易失性存储器匹配的数据流。“部分地伪无限”数据流被定义为将占据正在处理数据流对象的计算系统的易失性存储器的至少二分之一的数据流。枚举模块 601 可以响应于来自外部模块的请求,只枚举数据流的一部分,诸如,例如,图 2 的数据部分 210、分析法部分 220,或视图部分 230。枚举模块 601 可以要求枚举从数据流的第一元素开始。另一方面,枚举模块 601 可以允许枚举从数据流的任何其他部分开始(即,可以允许枚举中间流),而无需首先从数据流的开始处进行枚举。

[0082] 数据流对象 600 能够基于与数据流的被请求的成员元素相关联的属性来标识数据流的被请求的部分。在该情况下,数据流对象可包括处理请求并标识数据流的具有被请求的属性的所有成员元素的逻辑。例如,请求可以是针对一个城市的在特定经度和纬度坐标从 1000 英尺的高度往下看可见的所有元素。另一方面,数据流对象还能够对对于成员元素的明示请求作出响应。

[0083] 作为示例,伪无限数据流可以是对真实或虚构的城市的描述,包括对该城市内的所有项目的描述,包括对这些项目中的每一个的细节的每个可以想象的级别的描述。该信息可能太多,难以一次性地全部在存储器中表示。如此,数据流对象 600 可以只提供呈现当前视图所需的相关信息。例如,如果城市正在 100 英里的高度被查看,或许只有城市的边界信息是相关的。如果城市的一部分正在 10 英里的高度被查看,或许只提供有关较大的对象(如机场、停车场、水库等等,并且只有在视图内的情况下)的信息。如果城市的一部分正在一英里的高度被查看,或许由数据流对象提供呈现在视图内的街道和建筑物所需的信息。如果城市的一部分正在一千英里的高度被查看,或许可以提供呈现街道的更多细节(即,车道的数量、街道的名称、箭头,等等),以及有关建筑物的更多细节(纹理和窗户位置等等)所需的信息。自然,当发生此放大操作时,信息涉及小得多的地理区。在一百英尺视图中,数据流可以提供关于灌木林、检查井、排水沟、梯子、自动售货机、人行横道等的信息。对于常规计算机来说,可能难以为整个城市将这种数据量保存在存储器中。

[0084] 作为另一个示例,伪无限数据流可以在字面上是无限的,如在不规则碎片形的情况下。不规则碎片形是用数学方法定义的,以便不管在不规则碎片形的一部分中放大多少,

始终都能看得见重复的形状。始终可以进一步,进一步地,无限地放大。相同的原理也适用于缩小。需要花无限的信息量来在字面上表示这样的不规则碎片形的几何形状。然而,数据流对象可能具有如何用数学方法来定义不规则碎片形的概念。如果数据流对象被要求以特定细节级别产生不规则碎片形,则数据流对象可以计算适用于该特定细节级别的数据,并提供该数据。

[0085] 如此,伪无限数据对象能够生成被请求的数据范围,无论是通过评估定义细节的表达式和/或通过访问数据流对象外部的数据。这样的范围是被提供到管道 201 的分析法部分 400 的分层数据的示例。在一个实施例中,用于与数据流对象进行通信的协议可以是相同的,不管数据流本身有多大。如此,作者可以对与较小数据流对象相关联的数据流对象测试模型。那么,一旦其他人对模型充满信心,作者可以只将数据流对象切换到与无限数据流、伪无限数据流或部分地伪无限数据流相关联的数据流对象,而无需改变模型本身。

[0086] 不管规范数据 340 的形式如何,规范数据 340 是作为来自数据部分 300 的输出数据,并作为分析法部分 400 的输入数据来提供的。规范数据可包括其中包括各种数据类型的字段。例如,字段可包括简单数据类型,如整数、浮点数字、字符串、向量、数组、集合、分层结构、文本、XML 文档、表、列表、SQL 数据库查询结果、BI(商务智能)立方查询结果、各种格式的诸如 2D 图形和 3D 视觉模型之类的图形信息,或者甚至这些各种数据类型的复杂组合。作为另一个优点,规范化过程能够规范化各种输入数据。此外,数据部分 300 能够接受的输入数据的种类也是可扩展的。这在如在本说明书中稍后所讨论的组合多个模型的情况下是有帮助的。

[0087] 图 4 示出了表示图 2 的管道 201 的分析法部分 220 的示例的分析法部分 400。数据部分 300 将规范化的数据 401 提供给数据-模型绑定组件 410。规范化的数据 401 可以具有任何规范化的形式,以及任意数量的参数,参数的形式和数量甚至可以在一段输入数据与另一段输入数据之间不同。然而,为了讨论,规范数据 401 具有字段 402A 到 402H,它们此处可以统称为“字段 402”。

[0088] 另一方面,分析法部分 400 包括许多模型参数 411。模型参数的类型和数量可以随着模型不同而不同。然而,为了讨论特定示例,将讨论包括模型参数 411A、411B、411C 和 411D 的模型参数 411。在一个实施例中,可以在不使用命令性编码的情况下声明性地定义模型参数的标识以及模型参数之间的分析关系。

[0089] 数据-模型绑定组件 410 在规范化的数据字段 402 和模型参数之间调停,由此在字段和参数之间提供绑定。在此情况下,将数据字段 402B 绑定到模型参数 411A,如由箭头 403A 所表示的。换言之,数据字段 402B 的值被用来填充模型参数 411A。同样,在此示例中,数据字段 402E 被绑定到模型参数 411B(如由箭头 403B 所表示的),而数据字段 402H 被绑定到模型参数 411C(如由箭头 403C 所表示的)。

[0090] 数据字段 402A、402C、402D、402F 和 402G 未被示为绑定到模型参数中的任何一个。这将强调,并非输入数据中的所有数据字段都始终需要被用作模型参数。在一个实施例中,这些数据字段中的一个或多个可以用来向数据-模型绑定组件 410 提供与来自规范化的数据(对于此规范化的数据,或者或许任何将来类似的规范化的数据)的哪些字段将被绑定到哪一个模型参数有关的指令。这表示可以提供给图 2 的分析法部分 220 的分析法数据 221 的类型的示例。可以按许多方式形成来自规范化的数据中的哪些数据字段被绑定到哪

些模型参数的定义。例如,绑定可以 1) 由作者在创作时显式地设置,2) 由用户在使用时显式地地设置(服从由作者施加的任何限制),3) 由创作组件 240 基于算法启发式法自动绑定,和/或 4) 当确定无法通过算法进行绑定时,由创作组件提示作者和/或用户指定绑定。如此,也可以作为模型逻辑本身的一部分来解决绑定。

[0091] 作者定义哪些数据字段被映射到哪些模型参数的能力给予作者能够使用作者定义模型参数用起来舒服的符号的巨大灵活性。例如,如果一个模型参数表示压力,则作者可以将该模型参数命名为“Pressure”或“P”或对作者有意义的任何其他符号。作者甚至可以重命名模型参数,在一个实施例中,这可能导致数据模型绑定组件 410 自动地更新以允许以前绑定到旧名称的模型参数的绑定被改为绑定到新名称的模型参数,从而保留所需绑定。这种用于绑定的机制还允许绑定在运行时被声明性地改变。

[0092] 模型参数 411D 被示为带有星号,以强调数据-模型绑定组件 410 未向模型参数 411D 分配值。因此,模型参数 411D 仍然未知。换言之,未向模型参数 411D 分配值。

[0093] 建模组件 420 执行许多功能。首先,建模组件 420 定义模型参数 411 之间的分析关系。分析关系 421 被分类为三个一般类别,包括方程 431、规则 432 和约束 433。然而,解算器的列表是可扩展的。例如,在一个实施例中,可以作为分析关系的一部分来包括一个或多个仿真,假设提供了对应的仿真引擎并将其注册为解算器。

[0094] 如此处所使用的术语“方程”与在数学领域中使用的该术语一致。

[0095] 如此处所使用的术语“规则”表示条件语句,其中如果满足了一个或多个条件(条件语句的条件(即“if”)部分),则将采取一个或多个动作(条件语句的结果(即“then”)部分)。如果在条件语句中表达一个或多个模型参数,或者在结果语句中表达了一个或多个模型参数,则规则被应用于这些模型参数。

[0096] 如此处所使用的术语“约束”表示限制被应用于一个或多个模型参数。例如,在城市规划模型中,特定房屋元素可被限于置于具有总体可能的分区目的地的子集的地图位置上。桥元素可被限于低于某一最大长度或一定数量的车道。

[0097] 熟悉该模型的作者可以提供适用于该模型的这些方程、规则和约束的表达式。在仿真的情况下,作者可以提供适当的仿真引擎,该仿真引擎提供模型参数之间的适当的仿真关系。建模组件 420 可以提供供作者提供方程、规则和约束的自然的符号表达式的机制。例如,热力学相关模型的作者可以简单地从热力学教科书复制和粘贴方程。将模型参数绑定到数据字段的能力允许作者使用该作者熟悉的任何符号(如在作者依赖的教科书中所使用的原样的符号)或作者希望使用的确切符号。

[0098] 在求解之前,建模组件 420 还标识将求解哪些模型参数(即,此后称为“输出模型变量”(单数情况下)、或“诸输出模型变量”(复数情况下)、或“(诸)输出模型变量”(一个或多个输出模型变量的情况下))。(诸)输出模型变量可以是未知参数,或者它们可以是已知模型参数,其中已知模型参数的值在求解操作中变化。在图 4 的示例中,在数据-模型绑定操作之后,模型参数 411A、411B 和 411C 是已知的,而模型参数 411D 是未知的。因此,未知模型参数 411D 可以是诸输出模型变量中的一个。作为替换或补充,已知模型参数 411A、411B 和 411C 中的一个或多个也可以是输出模型变量。然后,如果可能的话,解算器 440 求解(诸)输出模型变量。在下面所描述的一个实施例中,解算器 440 能够求解各种输出模型变量,甚至在单个模型内,只要提供足够的输入模型变量以允许求解操作得以执行。输入

模型变量可以是,例如,其值在求解操作过程中不会变化的已知模型参数。例如,在图 4 中,如果模型参数 411A 和 411D 是输入模型变量,则解算器可以改为求解输出模型变量 411B 和 411C。在一个实施例中,解算器可以输出单个模型参数的许多不同数据类型中的任何一个。例如,不管操作数是整数、浮点、向量、或矩阵,某些方程运算(如加法、减法等等)都适用。

[0099] 虽然无论如何不是优选实施例,但是,有一个其中解算器 440 是通过电子表格程序来实现的实施例,电子表格程序中有多个电子表格,无论那些多个电子表格是位于单个电子表格文件中的多个工作表中和/或还是那些多个电子表格位于不同的电子表格文件中。在典型的电子表格中,有多个单元格。每一单元格都可以具有文字值或可以被解析成文字值的相关联的表达式。如果表达式可能依赖于来自其他单元格的值,则这些值也可能已通过与其他单元格相关联的相对应的表达式来解析。

[0100] 当在一个方向求解时,电子表格是有效的,其中,输入模型参数和输出模型参数是已知的。然而,归属于解算器 440 的优点之一是,取决于哪些模型参数被标识为输入参数、哪些模型参数被标识为输出模型参数而允许不同的求解方向,求解方向或许随着输入模型参数和/或输出模型参数的身份的变化而从一个求解方向变为下一个方向。这可以通过为每一个可能的求解方向分配不同的电子表格在电子表格程序中得以解决。例如,如果有二十个可能的求解方向,则总共可以有 20 个电子表格,每一个求解方向有一个电子表格。

[0101] 每一电子表格都具有适当地链接的单元格,带有用于在该方向求解的适当的表达式。在此实施例中,电子表格程序内部的或者外部的宏或其他可执行程序可以执行为建模组件 420 所描述的确定哪些参数是输入参数以及哪些参数将被求解的功能,在给定该求解方向的情况下,选择适当的电子表格,并为所选求解方向,填充适当的电子表格字段。一旦适当的输入参数字段被填充,电子表格将求解输出模型参数,并使用电子表格中的链接的表达式,将值填充到适当的输出模型参数字段。然而,回想一下,电子表格实现不是实施此处所描述的原理的优选实施例。例如,如果有数百个可能的求解方向,则在一个电子表格专用于每一求解方向的实施例中可以有数百个电子表格。如此,如果分析法将在此电子表格实施例中变化,则此电子表格实施例将涉及手动筛选每一个电子表格以查看应该如何改变链接的分析表达式。本描述将更远离特定电子表格实施例,并再一次进入对解算器 400 功能的一般讨论。

[0102] 在一个实施例中,甚至在解算器 440 不能求解出特定输出模型变量的情况下,解算器 400 仍可以呈现该输出模型变量的部分解,即使对于实际数值结果(或任何所求解的数据类型)的完全解是不可能的。这允许管道通过提示作者需要什么信息才能取得完全解来促进增量开发。这还有助于消除创作时和使用时之间的区别,因为在整个各种创作阶段至少部分解是可用的。举一抽象示例,假设分析法模型包括方程 $a = b + c + d$ 。现在假设 a , c 和 d 是输出模型变量,而 b 是具有已知值 5(在此情况下,整数)的输入模型变量。在求解过程中,解算器 440 只能够求解输出模型变量中的一个“ d ”,并将值 6(整数)分配给叫做“ d ”的模型参数,但是解算器 440 不能求解“ c ”。因为“ a ”依赖于“ c ”,叫做“ a ”的模型参数还仍是未知的,并未被求解。在此情况下,代替将整数值分配给“ a ”,解算器可以求部分解,并将串值“ $c+11$ ”输出到模型参数“ a ”。如上文所提及的,当领域专家正在创作分析法模型时,这可能特别有帮助,并在本质上用于提供和模型参数“ a ”的内容有关的部分信息,以及将用于提示作者需要提供允许求解“ c ”模型参数的某个进一步的模型分析法。此部分解结

果可以在视图合成中以某种方式输出,以允许领域专家看到该部分结果。

[0103] 在图 4 中以简化形式示出了解算器 440。然而,解算器 440 可以指导多个组成解算器的操作,如参考图 10 到 16 所描述的。在图 4 中,建模组件 420 随后使模型参数 411(包括现在已知的和已求解的输出模型变量)可用作输出以提供给图 5 的视图部分 500。

[0104] 图 5 示出了表示图 2 的视图部分 230 的示例的视图部分 500。视图部分 500 从图 4 的分析法部分 400 接收模型参数 411。视图部分还包括包含视图组件集合的视图组件储存库 520。例如,此示例中的视图组件储存库 520 被示为包括视图组件 521 到 524,但视图组件储存库 520 可以包含任意数量的视图组件。每一个视图组件都可包括零个或更多个输入参数。例如,视图组件 521 不包括任何输入参数。然而,视图组件 522 包括两个输入参数 542A 和 542B。视图组件 523 包括一个输入参数 543,且视图组件 524 包括一个输入参数 544。也就是说,这只是一个示例。输入参数可以,但是不必影响呈现视觉项目的方式。视图组件 521 不包括任何输入参数的事实强调了可以有不参考任何模型参数而生成的视图。考虑只包括不变的固定(内置)数据的视图。这样的视图可以例如构成用户的参考信息。可另选地,考虑只提供浏览目录的一种方式的视图,以便可以从其中选择项目以便导入到模型中。

[0105] 每一视图组件 521 到 524 都包括对应的逻辑或与对应的逻辑相关联,当由视图合成组件 540 使用对应的视图组件输入参数(如果有的话)来执行该逻辑时,使得对应的视图项目被置于虚拟空间 550 中。该虚拟项目可以是静态图像或对象,或者可以是动态的动画虚拟项目或对象。例如,视图组件 521 到 524 中的每一个都与对应的逻辑 531 到 534 相关联,当这些逻辑被执行时使得对应的虚拟项目 551 到 554 分别被呈现在虚拟空间 550 中。虚拟项目被示为简单形状。然而,虚拟项目在形式上可能十分复杂,或许甚至包括动画。在此描述中,当视图项目被呈现在虚拟空间时,这意味着视图合成组件已经创作了足够的指令,当这些指令被提供给呈现引擎时,呈现引擎能够在显示器上的指定位置并以指定的方式显示该视图项目。

[0106] 通过使用例如图 2 的创作组件 240,视图组件 521 到 524 或许甚至作为视图数据提供给视图部分 500。如上文参考图 6 所描述的,视图数据甚至可以由数据流对象所提供的伪无限或部分地伪无限的数据流的范围。例如,创作组件 240 可以提供允许作者从多个几何形状中选择或者可能构造其他几何形状的选择器。作者也可以指定每一个视图组件的输入参数的类型,而一些输入参数可以是视图部分 500 使用的默认输入参数。也可以给与每一视图组件 521 到 524 相关联的逻辑提供视图数据,和/或该逻辑也可以包括由视图部分 500 本身所提供的一些默认功能。

[0107] 视图部分 500 包括被配置成将至少一些模型参数绑定到视图组件 521 到 524 的对应的输入参数的模型-视图绑定组件 510。例如,模型参数 411A 被绑定到视图组件 522 的输入参数 542A,如箭头 511A 所示。模型参数 411B 被绑定到视图组件 522 的输入参数 542B,如箭头 511B 所示。同样,模型参数 411D 被分别绑定到视图组件 523 和 524 的输入参数 543 和 544,如箭头 511C 所示。模型参数 411C 未被示为绑定到任何对应的视图组件参数,强调了并非所有的模型参数都需要被管道的视图部分使用,即使那些模型参数在分析法部分是必需的。同样,模型参数 411D 被示为绑定到视图组件的两个不同的输入参数,表示模型参数可以被绑定到多个视图组件参数。在一个实施例中,模型参数和视图组件参数之间的绑定的定义可以通过下列方式制定:1) 由作者在创作时显式地设置,2) 由用户在使用时显式

地地设置（服从由作者施加的任何限制），3）由创作组件 240 基于算法启发式法自动绑定，和 / 或 4）当确定无法通过算法进行绑定时，由创作组件提示作者和 / 或用户指定绑定。

[0108] 再一次，尽管不是首选的，某些或全部视图部分 500 可以通过电子表格来实现。例如，单个电子表格可以充当一个或多个视图组件的基础，视图组件的相应的输入参数被在相对应的电子表格单元格中表示。视图组件的相关联的视图构造逻辑可以至少部分地使用电子表格程序内的链接的表达式来表示。然后，可以使用电子表格程序、宏或某种其他可执行程序的呈现功能来完成相对应的视觉项目的呈现。然而，如上文所提及的，这种基于电子表格的实现不是优选实施例，如此，此描述将回到视图部分 500 的更一般的实施例。

[0109] 如上文所提及的，视图项目可包括动画。举一个简单示例，考虑例如绘制了一个公司在给定时间点（如给定日历季度）的历史和计划收入、广告费用以及按销售区域的利润的柱状图。可以在所需时间跨度内为每一日历季度绘制柱状图。现在，想象您绘制这些图表中的一个，比方说时间跨度中的最早时间的那一个，然后每隔半秒将它替换为下一时间跨度的图表（例如，下一季度）。结果将是随着动画的进行，表示每一个区域的利润、销售以及广告费的柱线的高度发生变化。在此示例中，每一个时段的图表都是动画中的“单元”，其中该单元示出了移动之间的瞬间，其中顺序示出的单元集合模仿移动。常规动画模型允许使用内置的硬编码的图表类型来随着时间的推移而产生动画。

[0110] 然而，相比之下，通过使用管道 201，可以使任何类型的视觉项目动画化，并且可以通过改变视觉组件的参数中的任何一个或任何组合来驱动该动画。返回到上面的柱状图示例，想象我们按广告费来动画化，而非按时间动画化。此动画中的每一“单元”是示出了对于给定广告费的值的随着时间的推移的销售和利润的柱状图。如此，随着广告费变化，柱线响应于广告费的变化而增长和收缩。

[0111] 动画数据显示的力量是它们使得什么参数对其他参数的变化最敏感变得非常显而易见，因为您立即看到响应于动画参数的变化，每一个参数的值如何快速地变化以及变化到什么程度。

[0112] 管道 201 还由于下列特征而在其动画能力方面与众不同：

[0113] 首先，可以通过模型的分析法来计算动画变量的步骤的顺序，对照只是预定义的范围内的固定步骤顺序。例如，在作为动画变量来改变广告费的示例中，想象所指定的将是“通过广告费来进行动画化，其中对于每一个步骤，广告费增大 5%”或“其中广告费是该步骤的总费用的 10%”。一个复杂得多的示例是“通过广告费来进行动画化，其中广告费被优化以便使销售随时间的变化率最大化。”换言之，解算器将确定随时间（即，对于诸如季度之类的每一个连续的时间段）的广告花费的一组步骤，以使得销售的增长率最大化。这里，用户大概不仅希望看到通过改变广告费可以使销售增长多快，而且希望了解实现此增长的每季度广告费的量（可以作为合成视图的一部分来绘制该值的序列）。

[0114] 其次，可以使任何类型的视觉项目动画化，而不只是传统的数据图表。例如，考虑喷气式发动机的计算机辅助设计（CAD）模型，该模型 1）将通过风速参数动画化，2）其中，涡轮机的旋转速度是风速的函数，并且 3）其中，涡轮机轴承的温度是风速的函数。喷气式发动机对涡轮叶片不受损或者轴承过热之前涡轮机可以旋转得多快具有限制。如此，在此动画中，我们期望随着风速改变，涡轮叶片和轴承的颜色应该从蓝色（安全）变为红色（临界）。“安全”和“临界”涡轮机 RPM 的值和轴承温度可以由该模型基于那些部件的物理特

征来计算。现在,随着动画在定义的范围内改变风速,我们看到涡轮叶片和轴承都改变了颜色。现在有趣的是注意哪一个首先到达临界,二者中的任何一个是否经受突然(失控)达到临界。这些类型的效果难以通过查看图表或查看图形的序列来辨别,但是在动画中立即变得显而易见。这只是通过任意参数(风速)使任意视觉项目(CAD 模型)动画化的一个示例,动画仍影响其他任意参数(涡轮机 RPM 和轴承温度)。可以根据要充当动画变量的任何所需的参数来使任何视觉项目的任何参数动画化。

[0115] 第三,可以在中间停止管道 201,以便可以由用户修改数据和参数,然后重新启动或恢复动画。如此,例如,在喷气式发动机示例中,如果在给定风速看到失控加热开始,则用户可以在失控时停止动画,修改某些发动机设计标准,诸如轴承或轴承表面材料的类型,并且随后继续动画以查看该改变的效果。

[0116] 如同此处所讨论的其他功能一样,可以由作者来定义动画,和/或让用户操纵以测试各种情形。例如,模型可以被创作成允许某些视觉项目由用户根据用户自己选定的参数和/或在用户选定的动画变量的数据范围内而动画化(包括指定计算出的范围应该是所希望的能力)。这样的动画也可以并排地显示,如在其他“假设分析(what-if)”比较显示中那样。例如,用户可以在带有将来的不同的流行利率或不同的广告费用斜率的两种情形下比较按时间动画化的销售和利润的随时间的动画。在喷气式发动机示例中,用户可以比较改变轴承设计之前和之后情况的发动机的动画。

[0117] 此时,将参考图 7 描述如何使用合成框架来实际构建视图合成的具体示例,图 7 示出了包括房间布局 701(在房间内布置了家具)并包括“风水”计 702 的视图合成的 3-D 呈现 700。提供此示例只是为了示出此处所描述的原理如何应用于任何任意的视图合成,而不管是什么领域。因此,图 7 的示例,以及此处所描述的任何其他示例视图合成,应该被严格地视为只是允许通过参考非限制性的具体示例来更全面地理解抽象概念的示例,而不是限定本发明的更广泛的范围。此处所描述的原理可以适用于构建无数种类的视图合成。尽管如此,参考具体示例可以阐明更广泛的抽象原理。

[0118] 图 8 示出了用于生成视图构造的方法 800 的流程图。方法 800 可以通过图 2 的管道环境 200 来执行,因而将频繁地参考图 2 的管道环境 200,以及参考图 3 到 5 来对其进行描述,图 3 到 5 各自示出了图 2 的管道的各特定部分。尽管可以执行方法 800 来构建任何视图合成,将参考图 7 的视图合成 700 来描述方法 800。方法 800 的一些动作可由图 2 的数据部分 210 来执行,并在图 8 的左栏在标题“数据”下列出。方法 800 的其他动作可由图 2 的分析法部分 220 来执行,并在图 8 的左侧第二栏在标题“分析法”下列出。该方法的其他动作可以由图 2 的视图部分 230 来执行,并在图 8 的右侧第二栏在标题“视图”下列出。动作中的一个动作可由呈现模块来执行,并在图 8 右栏中在标题“其他”下列出。

[0119] 参考图 8,数据部分访问至少共同地影响将显示什么视觉项目或如何显示视觉项目中的给定的一个或多个视觉项目的输入数据(动作 811)。例如,参考图 7,输入数据可包括家具的每一个项目的视图组件。例如,床、椅子、植物、桌子、花中的每一个,甚至房间本身都可以由对应的视图组件来表示。视图组件可以具有适于该视图组件的输入参数。例如,如果使用动画,则一些输入参数可能影响动画的流动。一些参数可能影响视觉项目的显示,而某些参数可能不影响。

[0120] 例如,房间本身可以是一视图组件。一些输入参数可包括房间的尺寸、房间的朝

向、墙壁颜色、墙壁纹理、地板颜色、地板类型、地板纹理、房间内的光源的位置和电源等等。也可能有不一定反映在此视图合成中,但是可能反映在房间组件的其他视图和使用中的房间参数。例如,房间参数可能具有用度、分以及秒经度和纬度表示的房间的位置。房间参数也可以包括房间组件的作者的标识,以及房间的平均租金。

[0121] 房间内的各种组件也可以由对应的参数化的视图组件来表示。例如,每一植物都可以配置有指定盆风格、盆颜色、盆尺寸、植物颜色、植物弹性、植物对日光的依赖性、植物的每日摄水量、植物每日的氧气产生量、植物位置等等的输入参数。同样,取决于正在显示的东西的性质,这些参数中的某些参数可能影响如何呈现显示的方式,而其他参数可能不影响。

[0122] “风水”计 702 也可以是视图组件。该计量表可包括诸如直径、该计量表的直径中所包含的楔子(wedge)的数量、文本颜色等等之类的输入参数。“风水”计的各种楔子也可以是视图组件。在该情况下,对视图组件的输入参数可以是标题(例如,水、山、雷、风、火、地、湖、天)、或许是出现在楔子中的图形、色调等等。

[0123] 分析法部分将输入数据绑定到模型参数(动作 821),确定输出模型变量(动作 822),并使用模型参数之间的模型专用的分析关系来求解输出模型变量(动作 823)。上文已经讨论了动作 821 的绑定操作,该操作基本上允许作者灵活地使用模型作者舒服的符号来定义模型分析法方程、规则和约束。参考图 10 到 16 所描述的更加复杂的解算器可以旨在求解输出模型变量(动作 823)。

[0124] 输出模型变量的标识可以在一个求解操作与下一求解操作之间不同。即使模型参数可以保持相同,哪些模型参数的标识是输出模型变量将取决于将绑定到特定模型参数的数据的可用性。这在允许用户在给定视图合成中执行“假设分析”方面具有显著暗示。

[0125] 例如,在图 7 的“风水”房间示例中,假设用户购买了一把新椅子放在他们的起居室。用户可以将房间的设计作为数据提供给管道。这可以通过用创作组件提示用户输入房间尺寸来促进,或许还提供允许用户选择虚拟家具以拖放到虚拟房间中的、实际家具被置于实际房间中的适当位置的选择工具。然后,用户可以选择可以被编辑的一件家具,使其具有用户所购买的新椅子的特征。然后,用户可以将该椅子拖放到房间中。“风水”计 702 将自动地更新。在此情况下,椅子的位置及其他属性将是输入模型变量,而“风水”分数将是输出模型变量。随着用户将虚拟椅子拖放到各个位置,“风水”计的“风水”分数将更新,如此,用户可以测试将虚拟椅子放在各个位置的“风水”结果。为避免用户必须将椅子拖放到每个可能的位置以看出哪一个位置提供最佳的“风水”,用户可以获得局部视觉线索(诸如,例如,梯度线或箭头),这些局部视觉线索告诉用户从其当前位置在特定方向移动椅子使得“风水”更好还是更差,以及好多少或差多少。

[0126] 然而,用户也可以进行在常规视图合成中未听说过的别的操作。用户可以实际改变输出模型变量。例如,用户可以指示“风水”计中的所需的“风水”分数,并将虚拟椅子的位置保留为输出模型变量。然后,解算器将求解输出模型变量,并提供椅子的将至少达到指定的“风水”分数的一个或多个建议位置。用户可以选择使多个参数成为输出模型变量,而系统可以提供对这些输出模型变量的多个解。这通过参考图 10 到 16 更详细地描述的复杂解算器来促进。

[0127] 返回到图 8,一旦求解了输出模型变量,则将模型参数绑定到参数化的视图组件的

输入参数（动作 831）。例如，在“风水”示例中，在求解了未知“风水”分数之后，将分数作为输入参数绑定到“风水”计视图组件，或者或许绑定到该“风水”计中所包含的适当的楔子。可另选地，如果“风水”分数是输入模型变量，则可以求解虚拟椅子的位置，并将其作为输入参数提供给椅子视图组件。

[0128] 现在将呈现简化示例，该示例示出了解算器如何重新排列方程并改变一个分析模型的所有被驱动的输入和输出模型变量的指定的原理。用户自己不必重新排列方程。该简化示例可能不能准确地表示“风水”规则，但是示出了其原理。假设房间的总“风水”（FS）（FSroom）等于椅子的 FS（FSchair）和植物的 FS（FSplant）。假设 FSchair 等于常数 A 乘以椅子与墙的距离 d。假设 FSplant 是常数 B。房间的总 FS 是 $FS_{room} = A \times d + B$ 。如果 d 是输入模型变量，则 FSroom 是输出模型变量，其显示在“风水”计上的值随用户改变椅子的位置而改变。现在假设用户点击“风水”计来使它成为输入模型变量，并将 d 转换到未知输出模型变量状态。在此情况下，解算器有效地并在内部将上面的公式重写为 $d = (FS_{room} - B) / A$ 。在该情况下，随着用户在“风水”计上改变所需值 FSroom，视图组件可以四处移动椅子，改变其与墙的距离 d。

[0129] 然后，视图部分通过使用输入参数（如果有的话）执行与该视图组件相关联的构建逻辑以构建视觉项目的视图（动作 832），来或许驱动视图合成中的视图项目的构建。然后，可以将该视图构造提供给呈现模块，然后该呈现模块使用该视图构造作为呈现指令（动作 841）。

[0130] 在一个实施例中，构建视图的过程被视为由解算器执行的数据转换。即，对于给定类型的视图（例如，考虑柱状图），存在着包括规则、方程以及约束的模型，该模型通过将输入数据转换为可显示的输出数据结构（称为景物图）来生成视图，该可显示的输出数据结构编码了所有低级几何形状以及呈现软件所需的相关联属性以驱动图形硬件。在柱状图示例中，输入数据将是例如将与诸如图表标题、轴标签等事物的属性一起绘制的数据系列。生成该柱线的模型将具有执行下列操作的规则、方程以及约束：1) 统计该数据系列包括多少条目以便确定要绘制多少柱线，2) 计算该数据系列横跨的范围（最小值，最大值），以便计算诸如每一个轴的比例和起始 / 最终值等事物，3) 基于前面计算出的缩放因子，计算数据系列中的每一个数据点的柱线的高度，4) 统计图表标题中有多少字符，以便计算标题的起始位置和大小，以便适当地定位该标题并使其居于图表的中心等等。总而言之，模型被设计为基于输入数据来计算一组几何形状，这些几何形状被安置在类型为“景物图”的分层数据结构内。换言之，景物图是模型基于输入数据来求解的输出变量。如此，作者可以使用该作者用来创作、定制以及合成任何类型的模型的同一个框架，来设计全新种类的视图、定制的现有视图，并将预先存在的视图合成为合成图。如此，不是程序员的作者可以在不编写新代码的情况下创建新的视图。

[0131] 返回到图 2，回想一下用户交互响应模块 250 检测用户何时与视图合成进行交互，并使得管道恰当地作出响应。图 9 示出了用于对用户与视图合成的交互作出响应的方法 900 的流程图。具体而言，用户交互响应模块确定管道的哪些组件应该执行进一步的工作以便重新生成该视图，还向管道组件提供表示用户交互或至少取决于用户交互的数据。在一个实施例中，这是通过在相反的（上行）视图 / 分析法 / 数据方向上运行并且平行于（下行）数据 / 分析法 / 视图管道的转换管道来完成的。

[0132] 交互作为事件被张贴到上行管道中。数据 / 分析法 / 视图管道中的每一个变换器都提供处理传入的交互数据的上行变换器。这些变换器可以是空（通路，其在路径之外得到优化）或者它们可以对要进一步向上游馈送的交互数据执行变换操作。这在以下方面提供了管道的正面的性能和响应性：1) 对上行转换没有影响的交互行为（诸如对源数据没有效果的视图操纵）可以在管道中的最适当的（最不上游）点加以处理，2) 中间变换器可以在最终来自更上游的变换器的最后更新之前、通过将启发式地确定的更新发回到下游来优化视图更新性能。例如，在接收到数据编辑交互时，视图级变换器可以使立即视图直接更新到视图的景物图中（对于它知道如何解释的编辑），最后的完整更新稍后来自其中实际编辑了源数据的上游数据变换器。

[0133] 当给定视图交互的语义具有到所需的底层数据编辑的非平凡映射时，中间变换器可以提供所需的上游映射。例如，拖动计算出的结果的图上一个点可能需要将计算多个源数据项的新值的倒退解，这些源数据项在图上馈送计算出的值。解算器级上游变换器将能够调用所需解，并将所需的数据编辑传播到上游。

[0134] 图 9 示出了用于对用户与视图构造的交互作出响应的方法 900 的流程图。在检测到用户与显示器上的视图合成的呈现进行交互之后（动作 901），首先确定该用户交互是否需要重新生成视图（判定框 902）。这可以通过呈现引擎引发一个由图 2 的用户交互响应组件 250 解释的事件来执行。如果该用户交互不需要重新生成视图（在判定框 902 中为“否”），则管道不执行重构视图的任何进一步的动作（动作 903），但呈现引擎本身可以对视图执行某种转换。这样的用户交互的示例可以是用户增大视图构造的呈现的对比度或旋转该视图构造。由于这些动作可以由呈现引擎本身来执行，所以管道不需要响应于该用户交互执行重构视图的任何操作。

[0135] 另一方面，如果确定用户交互的类型需要重新生成视图构造（在判定框 902 中为“是”），则由管道重构视图（动作 904）。这可能涉及对提供给管道的数据的某种改变。例如，在“风水”示例中，假设用户将在虚拟房间内移动虚拟椅子的位置，则虚拟椅子组件的位置参数因而将变化。将引发一个事件，向分析法部分通知表示虚拟椅子的位置的对应的模型参数也应该改变。然后，分析法组件将重新求解“风水”分数，重新填充“风水”计或楔子的对应的输入参数，使得“风水”计利用适于椅子的新位置的当前“风水”分数来更新。

[0136] 用户交互可能需要以前已知的模型参数现在是未知数，以前未知的参数现在是已知的。这是可能要求输入和输出模型变量的指定发生变化以便以前指定的输入模型变量可能变为输出模型变量以及反之亦然的一个可能的示例中的一个。在该情况下，分析法部分将求解新输出模型变量，从而驱动视图合成的重构。

[0137] 这种类型的重构在构建由不同的数据驱动的视图合成的替换视图时是有帮助的。然而，这在说故事中也是有帮助的，通过导致在视图中从一个视图合成变换到下一视图合成。传统上，这种故事讲述是通过以下完成的：拍摄可视化内的数据的多个不同的配置的快照（意味着可视化也是不同的），然后，将这些快照变为幻灯片，在幻灯片之间有简单的变换。

[0138] 然而，这种方法对于比较复杂的视觉不好使，诸如或许“风水”示例中的房间的可视化，或者或许诸如牵引车之类的机器的复杂片段的可视化。与图表中所使用的简单可视化不同，在现实世界或其他复杂的可视化中，有什么可以被改变的大量置换。此外，在视觉

形象的可能的变化内有连续（例如，牵引车臂可以连续地移动到多个位置，该臂移动可以导致许多其他元素的重新定位（可能连续地）。或者或许，任何数量的家具或其他房间元素可以被来回移动并换掉。

[0139] 可以采取的另一种方法是将可视化脚本化为故事画板。这里的麻烦是，很难使脚本成为数据驱动的，脚本也难以以遵守数据和视觉形象内的约束及其他关系的方式、预示和允许用户可以进行的对数据和视觉形象的许多更改，以及随后的传播的对视觉形象的更改。

[0140] 另一方面，当使用此处所描述的管道 201 构建时，可以构建非常复杂的视图合成。每一个视图合成都可以是复杂场景的故事画板中的一个场景。故事画板可以通过以各种可能的方式中的一种方式来改变视图合成，从一个场景变换到另一个场景。例如，变换可以涉及下列变换中的一个、某些或全部：

[0141] 1) 视觉形象变换：在此变换中，驱动视图合成的数据可以保持相同，但是，用来构建视图合成的视图组件组可以变化。换言之，数据上的视图会变化。例如，一组数据可包括环境数据，包括温度、风速，及表示诸如雷击之类的环境活动的序列的其他数据。一个场景可以在飞机穿过该环境的上下文中表明该数据。故事画板中的下一场景可以讲述经受完全相同的环境条件的另一架飞机。

[0142] 2) 数据变换：这里，视图组件保持相同。换言之，使视图相同，但是，数据会影响视觉属性变化，从而改变视图被呈现的方式。例如，数据可以变化，和 / 或数据与模型参数的绑定会变化。

[0143] 3) 坐标系统变换。这里，数据以及视图组件组可以保持相同，但是，坐标系统从一个坐标系统变为另一个坐标系统。

[0144] 4) 目标世界变换。这里，一切可以保持相同，但是目标虚拟世界变化。例如，一个几何形状可以叠加在另一个几何形状之上。下面将提供几何形状，以及叠加的几何形状的示例。

[0145] 解算器框架

[0146] 图 10 示出了可以表示图 4 的解算器 440 的示例的解算器环境 1000。解算器环境 1000 可以用软件、硬件或其组合来实现。解算器环境 1000 包括管理和协调专门解算器的集合 1010 的操作的解算器框架 1001。集合 1010 被示为包括三个专门解算器 1011、1012 和 1013，但是省略号 1014 表示也可以有其他数量（即，三个以上或少于三个）的专门解算器。此外，省略号 1014 还表示专门解算器的集合 1010 是可扩展的。随着发现和 / 或开发可以帮助模型分析法的新的专门解算器，可以将这些新专门解算器合并到集合 1010 中，以补充现有专门解算器，或者或许替换现有解算器中的一个或多个。例如，图 10 示出了新解算器 1015 使用解算器注册模块 1021 被注册到集合 1010 中。作为一个示例，新解算器或许可以是接受一个或多个已知值并求解一个或多个未知值的仿真解算器。其他示例包括用于线性方程组、微分方程、多项式、积分、求根器、因式分解器、优化器等等的系统的解算器。每个解算器都可以在数值模式或在符号模式或在混合数值 - 符号模式下工作。解的数值部分可以驱动参数化的呈现下游。解的符号部分可以驱动部分解呈现。

[0147] 专门解算器的集合可包括适于求解输出模型变量的任何解算器。如果例如模型要确定自行车的制动器，则可以保证对复变微积分方程的求解。在该情况下，可以将专门复变

微积分解算器集成到集合 1010 中,以或许补充或替换现有方程解算器。在一个实施例中,每一个解算器都被设计成求解成特定类型的分析法关系中的一个或多个输出模型变量。例如,可以有被配置成求解一个方程中的未知数的一个或多个方程解算器。可以有被配置成应用规则以求解未知数的一个或多个规则解算器。可以有被配置成应用约束以求解未知数的一个或多个约束解算器。其他类型的解算器可以是例如使用输入数据执行仿真以便构建对应的输出数据的仿真解算器。

[0148] 解算器框架 1001 被配置成协调集合 1010 中的一个或多个或全部专门解算器的处理以使得一个或多个输出模型变量被求解。解算器框架 1001 被配置成将已求解的值提供给一个或多个其他外部组件。例如,参考图 2,解算器框架 1001 可以将模型参数值提供给管道的视图部分 230,使求解操作影响由视图组件执行来呈现视图项目的方式,或从而影响与视图项目相关联的其他数据。作为求解的另一潜在效果,可以改变模型分析法本身。例如,仅仅作为可以实现这一点的许多示例中的一个示例,可以利用可修改的规则集来创作模型,以便在给定解中,最初不活动的某些规则和 / 或约束被激活,并且最初被激活的某些规则和 / 或约束变得不活动。也可以用这种方法修改方程。

[0149] 图 11 示出了解算器框架 1001 协调集合 1010 中的专门解算器之间的处理的方法 1100 的流程图。现在将频繁地参考图 10 的解算器环境 1000 描述图 11 的方法 1100。

[0150] 解算器框架通过标识哪些模型参数是输入模型变量 (动作 1101),以及哪些模型参数是输出模型变量 (动作 1102),并通过标识定义模型参数之间的关系的模型分析法 (动作 1103),来开始求解操作。给定此信息,解算器框架分析模型参数中的依赖性 (动作 1104)。甚至给定固定一组模型参数,并给定固定一组模型分析法,依赖性也可以根据哪些模型参数是输入模型变量以及哪些是输出模型变量而变化。因此,每当使用哪些模型参数的标识是输入并基于模型分析法来执行求解操作时,系统可以推断依赖图。用户不必为每一个求解指定依赖图。通过评估每个求解操作的依赖性,解算器框架具有在一个求解操作过程中求解一组一个或多个模型变量并在下一求解操作中求解另一组一个或多个模型变量的灵活性。在图 2 到 5 的上下文中,这意味着用户通过与视图合成进行接口来指定输入什么以及输出什么的更大的灵活性。

[0151] 在某些求解操作中,模型可能根本没有任何输出模型变量。在该情况下,求解将验证全部已知模型参数值一并考虑时满足由该模型的分析法所表达的所有关系。换言之,如果要擦除任何一个数据值,将它变为未知数,然后求解,则被擦除的值将由模型来重新计算,并将与以前的相同。如此,加载的模型已经以求解的形式存在,并且当然,具有未知数并获得解的模型现在也以求解的形式存在。重要的是,与求解的模型的视图进行交互的用户能够编辑视图,这可以具有改变一个或多个数据值的效果,并且因此造成将尝试重新计算输出模型变量的数据值以便新的数据值集合与分析法相一致的重新求解。用户可以编辑哪些数据值 (不管模型是否以输出模型变量开始) 是由作者控制的;事实上,这是通过由定义哪些变量表示允许的未知数的作者来控制的。

[0152] 如果存在具有可以独立地被求解而无需首先求解其他表达式中的其他未知数的一个或多个未知数的表达式 (判定框 1105 中为“是”),则可以在任何时间甚至与其他求解步骤平行地求解这些表达式 (动作 1106)。另一方面,如果存在具有在不首先求解另一表达式中的未知数的情况下无法求解的未知数的表达式,则已经找到求解依赖性。在该情况

下,该表达式变为定义相对于另一表达式的操作的特定次序的关系结构(如依赖树)的一部分。

[0153] 在具有对其他表达式的互连的求解依赖性的表达式的情况下,基于所分析的依赖性来确定专门解算器的执行次序(动作 1107)。然后,解算器按确定的次序执行(动作 1108)。在一个示例中,在模型分析法被表达为方程、约束以及规则的情况下,执行的次序可以如下所示:1) 将带有依赖性的或作为独立表达式不完全可解的方程重写为约束,2) 求解约束,3) 求解方程,以及 4) 求解规则。规则求解可能导致数据被更新。

[0154] 一旦解算器按指定的次序被执行,则确定求解是否应该停止(判定框 1109)。如果例如求解了全部输出模型变量,或者如果确定尽管并非求解了所有输出模型变量,但是专门解算器不能执行任何操作以进一步求解更多输出模型变量,则求解过程应该停止。如果求解过程不应该结束(判定框 1109 中为“否”),则过程返回到对依赖性的分析(动作 1104)。然而,此时,输入和输出模型变量的标识可能由于被求解的一个或多个输出模型变量而发生变化。另一方面,如果求解过程应该结束(在判断框 1109 中为“是”),则求解结束(动作 1110)。然而,如果由于有太多的输出模型变量而导致模型无法被完全求解,则模型仍然可能在生成部分解时取得成功,其中给输出模型变量分配了反映求解能够进行得多么远的符号值。例如,如果模型具有方程 $A = B + C$, B 已知是“2”,是输入模型变量,但是 C 是输出模型变量, A 也是输出模型变量并需要被求解,则模型解算器不能产生 A 的数值,因为 B 是已知的,而 C 是未知的;如此,代替全解,解算器返回“ $2 + C$ ”作为 A 的值。如此,对于作者来说,什么附加变量需要变成已知的是显而易见的,无论通过给它提供一个值或通过添加可以成功地从其他输入数据产生所需的值的进一步的规则/方程/约束或者仿真。

[0155] 每当解算器框架检测到已知模型参数中的任何一个的值有变化,和/或每当解算器框架确定已知和未知模型参数的标识已经变化,则可以重复此方法 1100。求解能以至少两种方式进行。首先,如果能以符号方式对模型进行完全求解(即,如果能通过算法重写所有方程、规则以及约束,以使得对于每一个未知数都存在可计算的表达式),则这就完成了,计算该模型。换言之,生成每一个未知数的数据值,和/或调整允许调整的数据值。作为第二可能的方式,如果模型无法以符号方式完全求解,则以符号方式部分地对它进行求解,并且然后确定是否可以使用一个或多个数值方法来达到所需的解。此外,执行优化步骤,以便甚至在第一种情况下,确定使用数值方法相对于执行符号求解方法是否是计算所需值的更快方式。虽然符号方法可以更快,但还是有这样的情况:符号求解可能执行太多的项重写和/或太多的重写规则搜索,以至于放弃此方法并使用数值方法求解会更快。

[0156] 图 12 示出了表示图 10 的解算器环境 1000 的示例的解算器环境 1200。在此情况下,解算器协调模块 1210 操作以接收输入模型变量 1201,并协调前向解算器 1221、符号解算器 1222(或“倒相器”)以及数字解算器 1223 的动作,以便生成模型变量 1202(包括输出模型变量)。前向解算器 1221、符号解算器 1222 和数字解算器 1223 是可以在图 10 的解算器集合 1010 中的解算器的示例。

[0157] 解算器协调模块 1210 维护具有相对应的模型变量的模型分析法的依赖关系图。对于每一个求解操作,解算器协调模块 1210 可以确定模型变量中的哪些是输入模型变量,而模型变量中的哪些是输出模型变量,如此,将被求解。

[0158] 前向解算器 1221 求解正确地呈现的模型分析法以便是前向可解的。例如,如果在

模型分析法中只有一个公式 $A = B + C$ ，并且如果 B 和 C 是输入模型变量，那么，可以通过将 B 和 C 的值插入到公式中使用前向求解，并且确定 A 的所产生的值，来求解 A 。

[0159] 符号解算器 1222 重写模型分析法，以便是前向可解的。例如，假设在公式 $A = B + C$ 中，变量 A 和 C 是输入变量，而变量 B 是要被求解的输出变量。在此情况下，在不首先颠倒模型分析法的情况下（在此情况下，颠倒公式），模型分析法不能被前向求解。因此，符号解算器 1222 将公式 $A = B + C$ 重写为 $B = A - C$ 。现在，可以由前向解算器 1221 对颠倒的公式进行前向求解，以便输入变量 A 和 C 被插入到公式 $B = A - C$ 中，以获取变量 B 的值。

[0160] 某些公式在数学上不是可逆的，或者至少还没有发现如何颠倒某些类型的公式。此外，即使公式是可逆的，或者已知如何颠倒公式，则符号解算器 1222 可以简单地不能颠倒公式。或者或许与诸如数字求解之类的其他求解方法相比，符号解算器 1222 颠倒公式效率低。因此，在模型分析法不适当地可逆的情况下（或者因为求逆是不可能的，未知的，或者符号解算器没有启用），提供数字解算器 1223 以使用模型分析法来求解模型分析法。

[0161] 解算器协调模块 1210 被配置成管理每一个求解操作。例如，图 13 示出了用于管理求解操作以便可以求解模型分析法的方法 1300 的流程图。方法 1300 可以由解算器环境 1200 在解算器协调模块 1210 的指示下进行管理。

[0162] 解算器协调模块 1210 标识模型分析法的模型变量中的哪一个（哪些）对于特定求解是输入变量，以及模型变量中的哪一个（哪些）对于特定求解是输出模型变量（动作 1301）。如果例如在图 4 中由数据 - 模型绑定器组件 410 定义了输入和输出模型变量，甚至在给定恒定的模型变量组的情况下，输入模型变量和输出模型变量的身份可以从一个求解操作变为下一个。因此，对求解操作的协调可以从一个求解操作变为下一个。例如，甚至在给定恒定的模型分析法组的情况下，取决于输入模型变量，前向求解可以对于一个求解操作足够，对颠倒的分析法的求逆和前向求解可以对于另一求解操作足够，并且或许数字求解可以对于再一个求解操作足够。

[0163] 然而，如果在管道 201 的分析法部分 220 的上下文中实现，甚至模型分析法可以随着模型分析法被形成或或许与如前面所描述的其他模型分析法相结合而改变。解算器环境 1200 可以通过每当有改变时标识输入和输出模型变量，通过解决任何更改的模型分析法，并适当地求解，来解决这些变化。

[0164] 对于每一个求解，一旦标识了输入和输出模型变量（动作 1301），解算器协调模块 1210 确定在给定输入模型变量的情况下，是否将执行对输出参数的前向求解，而无需首先颠倒模型分析法（判断框 1302）。如果将执行前向求解（在判断框 1302 中，是），则使前向解算器 1221 对模型分析法进行前向求解（动作 1303）。此前向求解可以是对整体模型分析法的，或者也可以只对模型分析法的一部分的。在后一种情况下，方法 1300 可以再一次执行，只是这一次带有包括在前向求解中被求解的模型变量的全套输入模型变量。

[0165] 如果确定至少在不首先颠倒模型分析法的情况下对于特定求解不执行对输出参数的前向求解（在判断框 1302 中为否），然后，确定对于特定求解是否将颠倒模型分析法，以便前向求解可以求解输出参数（判断框 1304）。如果模型分析法（或模型分析法的至少一部分）将被颠倒（在判断框 1304 中是），则由符号解算器颠倒模型分析法（动作 1305）。此后，可以使用前向求解来求解被颠倒的模型分析法（动作 1303）。再一次，如果只有模型分析法的一部分以此方式被求解，则方法 1300 可以再次执行，但是利用扩展的输入模型变

量组。

[0166] 如果确定对于特定求解不颠倒模型分析法（在判断框 1304 中为否），然后，数字解算器可以使用数值方法来求解输出变量（动作 1306）。再一次，如果只有模型分析法的一部分以此方式被求解，则方法 1300 可以再次执行，但是利用扩展的输入模型变量组。

[0167] 因此，已经描述了灵活的解算器环境 1300，其中，可以求解各种模型分析法，不管哪些模型变量是输入的，哪些模型变量从一个求解操作输出到下一求解操作。

[0168] 图 14 示出了可以由如图 10 所示的解算器框架 1001 执行的另一方法 1400 的流程图。具体而言，解算器求解模型变量（动作 1401），该模型变量可以定义视图合成的视图组件的属性。例如，解算器可以使用已知的模型变量来求解未知的模型变量。在某些情况下，已知的模型变量可以是由另一个解算器所提供的输出，诸如当解算器构成上文参考图 10 和 11 所提及的诸如依赖树之类的关系结构的一部分。

[0169] 求解操作导致规范数据中的实际变化（动作 1402）。在解算器求解模型变量之后，将视图合成的视图组件的属性设置为已求解的模型变量的值（动作 1403）。例如，可以作为如图 5 所示的模型参数 411 的一部分提供已求解的模型变量，模型参数 411 可以绑定到第一视图组件 520 的输入参数 542。在某些情况下，被用来求解已求解的模型变量的已知的模型变量可以定义第一视图组件 520 的另一个属性。在这样的情况下，已知的模型变量和已求解的模型变量可以被绑定到第一视图组件 520 的各种输入参数 542。在其他情况下，已知的模型变量可以定义第二视图组件 520 的属性，诸如当第一视图组件是第二视图组件的子或父时。在这些其他情况下，已求解的模型变量可以被绑定到第一视图组件 520 的输入参数 542，而已知的模型变量可以被绑定到第二视图组件 520 的输入参数 542。

[0170] 在视图组件的属性被设置为已求解的模型变量的值之后，呈现包括视图组件的视图合成（动作 1404）。

[0171] 在需要时，有各种环境可以与方法 1400 一起使用。例如，图 15 示出了可以与方法 1400 一起使用的并可以以软件、硬件或组合来实现的环境 1500。具体而言，环境 1500 可包括可以构成属性 - 设置器 1502a、1502b、1502c 的一部分和 / 或由它们调用的一个或多个解算器 1501a、1501b、1501c。解算器 1501 可以被配置成求解未知的模型变量（图 14 的动作 1401），例如，通过使用已知模型变量来求解未知的模型变量。另外，属性 - 设置器 1502 可以将视图组件的属性设置为已求解的模型变量的值（图 14 的动作 1403）。然而，属性 - 设置器 1502 不必包括解算器 1501，并可以简单地接收已知的模型变量，然后，将视图组件的属性设置为接收到的模型变量的值。

[0172] 图 16 示出了可以由如图 15 所示的环境 1500 执行的另一方法 1600 的流程图。更详细地，调用环境 1500 的第一属性 - 设置器 1502a（动作 1601）。如果需要，可以由如图 5 所示的模型 - 视图绑定组件 510 调用第一属性 - 设置器 1502a 和其他属性 - 设置器 1502，和 / 或，第一属性 - 设置器 1502a 和其他属性 - 设置器 1502 构成模型 - 视图绑定组件 510 的一部分。

[0173] 在被调用之后，第一属性 - 设置器 1502a 设置视图合成的视图组件的第一属性（动作 1602）。例如，第一属性 - 设置器 1502a 可以将第一属性设置为模型变量 1503a 的值。在某些情况下，第一属性 - 设置器 1502a 可以简单地接收已知的模型变量 1503a，然后将第一属性设置为接收到的模型变量的值。在其他情况下，第一属性 - 设置器 1502a 可以调用

解算器 1501a, 该解算器 1501a 可以求解模型变量 1503a, 然后, 可以将第一属性设置为已求解的模型变量的值。

[0174] 除设置第一属性之外, 第一属性 - 设置器 1502a 也调用第二属性 - 设置器 1502b (动作 1603)。然后, 第二属性 - 设置器 1502b 调用可以被配置成求解模型变量的诸如解算器 1501b 之类的解算器 (动作 1604)。具体而言, 如图 15 所示, 解算器 1501b 可以由第二属性 - 设置器 1502b 调用, 和 / 或解算器 1501b 构成第二属性 - 设置器 1502b 的一部分, 并且当被调用时, 解算器 1501b 可以求解未知的模型变量 1503b。在某些情况下, 解算器 1501b 可以通过使用已知的模型变量, 例如, 模型变量 1503a, 来求解未知的模型变量 1503b。例如, 当第一属性 - 设置器 1502a 调用第二属性 - 设置器 1502b (动作 1603) 时, 第一属性 - 设置器 1502a 可以将模型变量 1503a 传递到第二属性 - 设置器 1502b。然后, 解算器 1501b 可以通过使用模型变量 1503a 来求解未知的模型变量 1503b。当然, 第一属性 - 设置器 1502a 不必将模型变量 1503a 传递到第二属性 - 设置器 1502b, 该第二属性 - 设置器 1502b 可以以任何其他合适的方式来访问模型变量 1503a。

[0175] 然后, 第二属性 - 设置器 1502b 将视图合成的视图组件的第二属性设置 (动作 1605) 为, 例如, 已求解的模型变量 1503b 的值。当然, 在某些情况下, 第二属性 - 设置器 1502b 可以简单地接收已知的模型变量 1503b, 然后, 第二属性 - 设置器可以将第二属性设置为接收到的模型变量的值。因此, 第二属性 - 设置器 1502b 不必调用解算器 (动作 1604)。在第二属性 - 设置器 1502b 设置第二属性之后, 呈现包括视图组件的视图合成 (动作 1606)。

[0176] 在某些情况下, 第一属性和第二属性可以是视图合成的单个视图组件的属性。因此, 单个视图组件的属性 - 设置器 1502a、1502b 可以将第一和第二属性设置 (动作 1602、1605) 为模型变量 1503a、1503b 的值, 如此, 允许模型变量 1503a、1503b 定义第一和第二属性。

[0177] 在其他情况下, 第一属性可以是第一视图组件的属性, 而第二属性可以是第二视图组件的属性。因此, 第一视图组件的属性 - 设置器 1502a 可以将第一属性设置 (动作 1602) 为模型变量 1503a 的值, 如此, 允许模型变量 1503a 定义第一属性。另外, 第二视图组件的属性 - 设置器 1502b 可以将第二属性设置 (动作 1605) 为模型变量 1503b 的值, 如此, 允许模型变量 1503b 定义第二属性。在需要时, 第一视图组件可以是第二视图组件的子或父, 而第二视图组件可以是第一视图组件的子或父。

[0178] 如此, 如上文参考图 11 所述, 可以按显式的顺序编写多个解算器 (动作 1107), 然后, 根据该顺序求解 (动作 1108)。例如, 可以使用诸如依赖树之类的关系结构来显式地编写解算器。另外, 如参考图 14 到 16 所示, 可以基于具有解算器 1501 的属性 - 设置器 1502 的调用其他具有解算器 1501 的属性 - 设置器 1502 的能力, 隐式地编写多个解算器。

[0179] 合成视图合成

[0180] 参考图 2, 管道环境 200 还包括或许被作为创作机制 240 的一部分来包括的模型导入机制 241。模型导入机制 241 向作者提供用户界面或其他帮助, 以允许作者将预先存在的分析法驱动模型的至少一部分导入到用户正在构建的当前分析法驱动模型中。因此, 当创作新分析法模型时, 作者不必始终从头开始。导入可以是整个分析法驱动模型, 或者或许是该模型的一部分。例如, 导入可以造成下列六个潜在的效果中的一个或多个或全部。

[0181] 作为导入的第一潜在的效果,可以将另外的模型输入数据添加到管道中。例如,参考图 2,可以向输入数据 211、分析法数据 221 和 / 或视图数据 231 中添加另外的数据。另外的模型输入数据也可以包括被添加到图 3 的数据访问组件 310 或者可能不同的规范化组件 330 的另外的连接器。

[0182] 作为导入的第二潜在效果,在模型输入数据和模型参数之间可以有另外的或经过修改的绑定。例如,参考图 4,数据 - 模型绑定器 410 可能使得在规范化的数据 401 和模型参数 411 之间发生另外的绑定。这可能使得已知模型参数的数量增加。

[0183] 作为导入的第三潜在效果,可以有用以生成补充的一组模型参数的另外的模型参数。例如,参考图 4,模型参数 411 可能由于导入的模型的分析行为的导入而扩充。

[0184] 作为导入的第四潜在效果,可以有添加到模型的另外的分析关系(如方程、规则和约束)。另外的输入数据由第一潜在效果产生,另外的绑定由第二潜在效果产生,另外的模型参数由第三潜在效果产生,而另外的分析关系由第四效果产生。这些另外的项目中的任何一个或多个可以被视为影响视图合成的另外的数据。此外,这些效果中的任何一个或多个都可以改变图 4 的解算器 440 的行为。

[0185] 作为导入的第五潜在效果,在视图的模型参数和输入参数之间可以有另外的或不同的绑定。例如,参考图 5,模型 - 视图绑定组件 510 将潜在地扩充的一组模型参数 411 绑定到视图组件储存库 520 中的潜在地扩充的一组视图组件。

[0186] 作为导入的第六潜在效果,可以有添加到图 5 的视图组件储存库 520 的另外的参数化的视图组件,或许造成新视图项目被添加到视图合成中。

[0187] 因此,通过导入另一模型的全部或一部分,导入与该模型相关联的数据。由于视图合成是数据驱动的,所以这意味着该模型的导入的部分被立即合并到当前视图合成中。

[0188] 当导入预先存在的分析法驱动的分析法模型的一部分时,被提供给管道 201 的数据发生变化,从而造成管道 201 立即或响应于某种其他事件重新生成视图合成。因此,在基本上是从现有模型进行的复制和粘贴操作的情况下,由于解析操作,所产生的合成模型可以立即在显示器上查看。

[0189] 作为此功能如何有用的示例,考虑图 7 的“风水”房间视图合成。此应用程序的作者可以是“风水”专家,并可能希望只从标准房间布局视图合成模型开始。因此,通过导入预先存在的房间布局模型,“风水”专家现在相对快速地,如果不是立即的话,能够看到房间布局 701 在显示器上显示出来,如图 7 所示。不仅如此,标准房间布局视图合成模型通常可能附带的家具和房间项目目录,现在已经变得对图 7 的“风水”应用程序可用。

[0190] 现在,“风水”专家可能希望导入基本的饼图元素作为用于构建“风水”图表元素 702 的基础。然而,“风水”专家可为该图表元素指定特定的固定输入参数,包括或许总共有 8 个楔子,或许还有每一个楔子的背景图像和标题。现在“风水”专家只需指定分析关系,该分析关系指定模型参数如何相互关联。具体而言,家具的颜色、位置以及类型或其他房间项目都可能对特定“风水”分数具有影响。专家可以简单地记下这些关系,以由此通过分析将房间布局 701 和“风水”分数互连。这种在其他人的工作上构建的协作能力在创建解决问题并允许视觉分析的应用程序时产生巨大的创造性。这与可能允许用户使用固定依赖图来可视地编程单向数据流的系统完全不同。这些系统可以进行单向求解,最初编程的从输入到输出的方向。给定与用户的交互会话,取决于在任何时间什么是已知的以及什么是未知

的,此处所描述的原理允许以多种方式求解。

[0191] 视觉交互

[0192] 直到此时视图合成过程被描述为一次呈现的单个视图合成。例如,图 7 示出了从一组输入数据生成的单个视图合成。然而,此处所描述的原理可以被扩展到其中包括多个组成视图合成的集成视图合成的示例。这可能在许多不同情况下有用。

[0193] 例如,给定单个一组输入数据,当解算器机制正在求解输出模型变量时,可以有多个可能的解。组成视图合成可能各自表示多个可能的解中的一个,其中另一组成视图合成可能表示另一可能的解。

[0194] 在另一示例中,用户可能简单地希望保留使用特定一组输入数据生成的以前的视图合成,并随后修改该输入数据来尝试新的情形以生成新视图合成。用户还可能希望也保留该第二视图合成,并通过再次改变该输入数据来尝试第三种可能的情形。然后,用户可以同时查看这三种情形,或许通过并排比较,以获取通过一次只查看一个视图合成可能难以获得的信息。

[0195] 图 17 示出了从图 7 的“风水”示例延伸的集成视图合成 1700。在该集成视图合成中,再次使用元素 701 和 702 来表示图 7 的第一视图合成 700,与图 7 完全相同。然而,这里,有强调地表示的第二视图合成。第二视图合成与第一视图合成的相似之处是有两个元素,房间显示和“风水”分数计。然而,第二视图合成的输入数据不同于第一视图合成的输入数据。例如,在此情况下,家具的多个项目的位置数据将不同,从而导致它们在第二视图合成的房间布局 1701 中的位置不同于第一视图合成的房间布局 701 中的位置。然而,与第一视图合成的“风水”计 702 相比,各种家具项目的不同位置与第二视图合成的“风水”计 1702 中的不同“风水”分数相互关联。

[0196] 集成视图合成也可以包括比较元素,该比较元素在视觉上表示至少一个参数的值跨全部以前创建的并且目前显示的视图合成的某一些的比较。例如,在图 13 中,可以有柱状图,该柱状图示出每一个显示的视图合成的成本和交付时间。这样的比较元素可以是视图组素储存库 520 中的另外的视图组件。或许,比较视图元素只有在显示多个视图合成时才呈现。在该情况下,可以将比较视图合成输入参数映射到该模型的不同求解迭代的模型参数。例如,可以将比较视图合成输入参数映射到为图 17 的第一和第二视图合成的生成而产生的成本参数,并将其映射到为第一和第二视图合成的生成而产生的交付参数。

[0197] 参考图 17,还有允许用户在视觉上强调全部可用的以前构建的视图合成的选定子集的选择机制 1710。选择机制 1710 被示为包括以缩略图形式示出的或以某种其他不强调的方式示出的三个可能的视图构造 1711、1712 和 1713。每一个缩略图视图合成 1711 到 1713 都包括对应的复选框 1721 到 1723。用户可以选中与将被在视觉上强调的任何视图合成相对应的复选框。在此情况下,复选框 1721 和 1723 被选中,从而使得显示对应的视图构造的更大形式。

[0198] 集成视图合成或者甚至该集成视图合成的任何单个视图合成,都可以具有用户与视图合成进行交互以指定什么模型参数应该被视为未知的从而触发通过分析解算器机制的另一求解的机制。例如,在图 17 的房间显示 1701 中,可以右击家具的特定项目,右击特定参数(例如,位置),并且可以出现允许用户指定该参数应该被视为未知数的下拉菜单。然后,用户可以右击协调性百分比(例如,“风水”分数计 1702 中的 95%),于是会出现一

个滑动块（或其他用户输入机制的文本框），该滑动块允许用户指定不同的协调性百分比。由于这将造成已知和未知参数的标识的改变，所以这将造成重新求解，其位置被指定为未知的家具的项目可能出现在新位置。

[0199] 交互视觉线索

[0200] 在一个实施例中，集成的视图合成还可包括与视觉项目相关联的视觉提示或线索。视觉线索向用户给予某种视觉指示：1) 可以与其进行交互的相关联的视觉项目，2) 对该视觉项目可能有什么类型的交互，3) 如果对视觉项目作出了特定交互，结果将是什么，和/或 4) 为了取得结果，与一个或多个其他视觉项目的交互是否是必需的。

[0201] 如上文参考图 5 所提及的，视图部分 500 包括视图储存库 520，该视图储存库 520 包括多个视图组件 521、522、523、524。视图组件 522、523 和 524 中的一些是由相对应的输入参数（视图组件 522 的参数 542A 和 542B，视图组件 523 的参数 543，以及视图组件 524 的参数 544）内填充的值驱动的。被提供到输入参数的数据驱动相对应的视图组件的执行逻辑，以便数据控制视觉项目的构造。例如，视觉项目 552 的结构可以取决于被提供到输入参数 542A 和 / 或输入参数 542B 的数据。

[0202] 在一个实施例中，给定视图组件的输入参数中的一个或多个可以是交互性参数。交互性参数可以定义视觉项目是否将被带有交互性地呈现。可另选地，或另外地，交互性参数可以导致特定类型的交互性被应用于作为执行相对应的视图组件的执行逻辑的结果构建的相对应的视觉项目。下面，包含至少一个这样的交互性参数的视图组件将被称为“在视觉上带线索的交互式”视图组件。除启用交互性之外，被提供到交互性参数的数据还可定义相对应的视觉项目在被呈现时在视觉上如何被提示，以在视觉上通知用户视觉项目是交互式的，以及潜在地，还有交互性的类型。所呈现的视觉项目中的一个、某些或者甚至全部都可以是交互式，并以此方式在视觉上被提示。

[0203] 有许多不同类型的可以由视觉项目提供的交互性。一个此处被称为导航交互性。当用户导航地与视觉项目进行交互时，被用来构建用于呈现的视觉项目的视图组件的子集会变化。

[0204] 例如，一种类型的导航交互此处被称为“辖域交互性”。辖域交互性改变呈现的视觉项目的子集，以便恰好在导航交互性之前显示的至少一些视觉项目也紧随在导航交互性之后显示。例如，参考图 5，虚拟空间 500 被示为包括四个视觉项目 551、552、553 和 554。如果视觉项目 552 具有辖域交互性，则用户可以与视觉项目 552 进行交互，以便视觉项目 551 和 554 不再在要被呈现的虚拟空间 550 中。然而，视觉项目 552 和 553 可以保持在虚拟空间 550 中。可另选地或另外地，进一步的视觉项目可以被构建到视觉空间中。

[0205] 一种类型的辖域交互性是改变被用来生成用于呈现的视觉项目的视图组件的范围的滚动交互性。图 18A 和 18B 表示滚动交互性的示例。图 18A 表示滚动交互性之前的虚拟空间 1800A。这里，视图合成组件（参见图 5）构建了六个视觉项目 1811、1812、1813、1814、1818 和 1816。视觉项目 1811 被用指示视觉项目已经启用了向左滚动交互性的视觉线索来装饰（如星号 1801 所示）。视觉项目 1816 被用指示视觉项目已经启用了向右滚动交互性的视觉线索来装饰（如星号 1899 所示）。图 18B 表示在用户与视觉项目 1816 进行交互以向右滚动之后的虚拟空间 1800B。这里，视觉项目 1811 被从虚拟空间去除。另外，新视觉项目 1817 被添加到虚拟空间。在编辑交互性的示例中，在此情况下，滚动交互性还导

致视觉项目 1812 被用表示启用了向左滚动交互性的视觉线索来装饰。此外,视觉项目 1816 还丢失被提供到视觉项目 1817 的其视觉线索和交互性。可以通过简单地改变被提供以填充相对应的视图组件的输入参数的数据,对于给定视觉项目启用和禁用视觉线索和交互性功能。

[0206] 另一种类型的辖域交互性是改变被用来生成用于呈现的视觉项目的视图组件的“细节”级别的“细节交互性”。“放大”交互性可能导致较细粒度的视觉项目的出现,或许一些以前的视觉项目从视图中消失。“缩小”交互性可以导致较粗粒度的视觉项目出现,或许一些以前的较细粒度的视觉项目消失。例如,如果在可见的宇宙的图中放大,则星系的簇可以开始出现。如果将在该星系簇上放大,则单个星系可以开始成形。如果放大那些单个星系中的一个,如银河系,则单个星可以出现。如果放大那些星中的一个,如太阳,则太阳的细节可能变得越来越明显,或许有行星开始出现。如果放大那些行星中的一个,如地球,则诸如洲之类的大规模的地貌可能出现。如果进一步放大,则国家边界可能出现,稍后城镇可能会出现,然后,街道可能成形。这可能持续,直到子原子粒子成形。这样的放大拓扑图的粒度的粗略度不必在物理上是相关联的,如在宇宙的模型中那样。可以利用每一个辖域交互性在其他拓扑图中导航,导致以前的视觉项目消失,并导致新的视觉项目出现。再一次,对伪无限数据系列的使用可以促进此操作。

[0207] 在详细地描述交互性操作之前,图 19A 示出了示例虚拟空间 1900A。这里,只有两个视觉项目 1911 和 1912 开始显示。如图 19B 的虚拟空间 1900B 所示,视觉项目 1911 被放大,现在某些视觉项目 1921 和 1922 被呈现在视觉项目 1911 内部,而视觉项目 1912 现在超出视图范围。这是放大交互性的示例。作为缩小交互性的示例,可以从图 19B 的虚拟空间 1900B 开始,虚拟空间 1900A 表示缩小交互性之后的状态。

[0208] 交互性的类型还可是导致至少一个呈现的帧(以及或许整个显示)显示与之前显示的视觉项目完全不同的视觉项目的链接交互性。例如,在上面的图 7 中的“风水”房间示例中,点击特定视觉项目(如“风水”计 702)可能导致关于“风水”的网页被显示,代替“风水”房间视图。可另选地,或许有如果与其进行交互可能导致完全不同的房间出现的视觉项目。

[0209] 再一个类型的交互性可以是导致独立于被呈现的视觉场景的某种动作被采取的外部动作交互性。例如,可以与可能导致电子邮件被发送或设置警告、预定数据备份、执行性能检查等等的视觉项目进行交互。

[0210] 交互性的类型还可是编辑交互性。编辑交互性以一个或多个视图组件的一个或多个输入参数值改变的方式来改变数据。如果该输入参数影响视觉项目被构建的方式,那么,作为结果,视觉项目也会改变。数据中的变化还可导致输入模型参数的值变化,或导致输入模型参数和/或输出模型参数的身份变化。如此,施加于视觉项目的编辑交互性可以导致对分析法部分 400 的整个的重新计算。现在将提供编辑交互性的多个示例。

[0211] 图 20 示出了连续的美国 2000 的呈现。可以从一个视图组件构建美国视觉项目。然而,可以从相对应的子视图组件构建组成的州。对应于的该州的视觉项目的高度表示州的一些参数(例如,评估中的特定产品的按人口计算的消费)。这里,新墨西哥州视觉项目 2001 具有面向上的箭头 2011 形式的视觉线索。这提示用户,可以与新墨西哥州视觉项目 2001 进行交互,以便可以改变视觉项目的高度。同样,内华达州视觉项目 2002 和佛罗里达

州视觉项目 2003 分别具有相对应的面向下的箭头 2012 和 2013。这些箭头可以在视觉上提示用户,在新墨西哥州箭头 2011 的高度的向上的调整在通过分析法部分 400 对模型重新分析之后,会导致在内华达州视觉项目 2002 和佛罗里达州视觉项目 2003 的高度的向下调整。可另选地或另外地,箭头 2012 和 2013 可以指示,如果内华达州视觉项目 2002 和佛罗里达州视觉项目 2003 的高度两者都由用户向下调整,那么,将向上调整新墨西哥州视觉项目 2001 的高度。为了确定特定用户交互的后果,管道 201 可以考虑用户如何可以与所呈现的视觉项目进行交互,并执行对分析模型的相依度求解,以确定后果将是什么。

[0212] 图 21 示出了包括相关联的柱状图 2110 和饼图 2120 的图表 2100。柱状图 2110 包括多个柱线。柱线 2111A 中的一个被示为带有箭头 2111B 以表示可以垂直地调整此高度。这可以导致在饼图 2120 中分配各种楔形图时进行某些调整。可以在视觉上合并柱状图 2110 和饼图 2120。例如,所产生的饼图可包括楔形图,其厚度取决于柱状图相关联的柱的高度。

[0213] 图 22 示出了飓风映射图表 2200,其中,绘制了飓风 2211 的多条路线 2201。视觉线索 2220 指示用户可以与路线 2201 进行交互,以便将路线 2101 改变为,例如,路线 2202。这允许用户评估飓风的路线的可能的替换的现实。控件 2221、2222、2223 和 2224 还允许飓风的各种参数被改变,诸如,例如,风速、温度、旋转,以及飓风迁移速度。

[0214] 例如,在图 7 的“风水”示例中,如果特定协调性分数被指定为已知输入参数,则可为家具的该项目(其位置被指定为未知)建议家具的各个位置。例如,或许从家具发出多个箭头,建议移动该家具以便获得较高的协调性百分比的方向,移动以最大化水分数的不同方向,移动以最大化水分数的不同方向等等。视图组件也可以示出椅子可以被移动到的阴影以增加特定分数。如此,用户可以使用这些视觉提示以便围绕需要被优化的特定参数来改进设计。在另一示例中,或许用户希望降低成本。然后,用户可以指定该成本作为要被最小化的未知数,造成不同的一组建议的家具选择。

[0215] 图 23 示出了用于与显示多个视觉项目的用户界面进行交互的方法的流程图。如前面所描述的,计算机在显示器上呈现数据驱动的视觉项目(动作 2301)。回想一下,通过将数据提供到参数化的视图组件,形成每一个数据驱动的视觉项目。该数据又可能通过分析模型响应于数据被提供到管道 201 的分析法部分 400,或响应于数据被提供到管道 201 的数据部分 300 来获取。另外,视觉项目中的一个、某些或全部具有向用户表达可以与视觉项目进行交互和/或交互性类型的视觉线索或其他视觉强调(动作 2302)。在一个实施例中,视觉线索首先只表示相对应的视觉项目是交互式的,直到用户选择视觉项目(通过利用指针悬停在视觉项目上),视觉线索才被修改或补充以表达交互性类型。

[0216] 然后,计算系统检测发生了用户和视觉项目之间的预定的物理交互(动作 2303)。作为响应,对于该特定视觉项目,启用或激活交互性(动作 2304)。适当的响应将取决于交互性是导航交互性、链接交互性还是编辑交互性。在编辑交互性的情况下,结果还将取决于如管道的分析法部分 400 所定义的各种视觉项目之间的分析关系。

[0217] 交互性可以是导航、链接,以及编辑交互性中的两个或更多的组合。例如,在图 20 的美国示例中,对新墨西哥州视觉项目 2001 的高度的向上调整可以导致对内华达州视觉项目 2002 和佛罗里达州视觉项目 2003 的高度的向下调整(表示编辑交互性的示例)。然而,这还可导致显示被拆分成四个帧,三个帧中的每一个放大三个州:内华达州、新墨西哥

州和佛罗里达州（辖域交互性的示例）。另外，第四帧可以包含和新墨西哥州的住宅的消费偏好有关的调查信息（链接交互性的示例）。

[0218] 图 24 抽象地示出了表示另一应用示例的用户界面 2400。在此应用示例中，描述了允许用户使用此处所描述的原理轻松地构建数据驱动的视觉场景的方便的用户界面。

[0219] 用户界面 2400 还包括第一组 2411 视觉项目。在此所示出的情况下，组 2411 只包括一个视觉项目 2411。然而，此处所描述的原理不仅限于视觉项目组 2411 内的一个视觉项目。视觉项目 2411 具有相关联的数据 2412，如此，有时被称为“数据视觉项目”2411。

[0220] 尽管并非必需，但是在此示例中，相关联的数据被细分为多个数据组。任何数量的组都将足够。然而，四个数据组 2413A、2413B、2413C 和 2413D 被示为包括在图 24 的相关联的数据 2412 内。在所示出的相关联的数据 2412 中，平行地组织每一个数据组，每一个都具有相关联的多个数据字段。在所示出的情况下，所示出的相关联的数据 2412 中的每一个数据组都包括相对应的字段 a、b 和 c。数据组 2413A 的字段 a、b 和 c 将分别被称为数据字段 2413Aa、2413Ab 和 2413Ac。数据组 2413B 的字段 a、b 和 c 将分别被称为数据字段 2413Ba、2413Bb 和 2413Bc。数据组 2413C 的字段 a、b 和 c 将分别被称为数据字段 2413Ca、2413Cb 和 2413Cc。最后，数据组 2413D 的字段 a、b 和 c 将分别被称为数据字段 2413Da、2413Db 和 2413Dc。

[0221] 用户界面 2400 还包括第二组 2420 视觉项目。在所示出的情况下，组 2420 包括三个相对应的视觉项目 2421、2422 和 2423。然而，对第二组 2420 中的视觉项目的数量没有限制，也没有第二组中的视觉项目是相同类型的任何要求。下面，视觉项目 2421、2422 和 2423 还可被称为“元素视觉项目”。每一个视觉项目 2421、2422 和 2423 可以，例如，通过使用输入参数来执行相应的视图组件的相对应的逻辑来构建。这样的视图组件可以对于视图组件 521 到 524 参考图 5 所描述的那些。如此，视觉项目 2421、2422 和 2423 中的每一个都被示为具有输入参数。这样的输入参数可以是提供到视图组件的输入参数，或者也可以是驱动视觉项目的呈现的输入参数。只作为示例，视觉项目 2421、2422 和 2423 中的每一个都被示为每一个都具有三个输入参数。具体而言，视觉项目 2421 被示为具有输入参数 2421a、2421b 和 2421c。视觉项目 2422 被示为具有输入参数 2422a、2422b 和 2422c。视觉项目 2423 被示为具有输入参数 2423a、2423b 和 2423c。

[0222] 用户界面 2400 还包括用户交互机制 2440。用户交互机制 2440 准许用户（通过一个或多个用户手势）导致数据视觉项目 2411 的数据 2412 应用于元素视觉项目 2420 的输入参数。在一个实施例中，用户手势可以实际导致相关联的数据被绑定到元素视觉项目 2420 的输入参数。这样的用户手势可以是拖放操作、悬停操作、拖动和点击，或任何其他用户手势或手势的组合。如此，用户可以将来自数据视觉项目 2410 的数据应用或绑定到元素视觉项目的输入参数，从而使用简单手势来改变元素视觉项目的外观。在一个实施例中，这甚至不涉及用户输入任何相关联的数据，并允许单个手势组将数据应用和 / 或绑定到多个元素视觉项目。下面将进一步描述将来自一个数据视觉项目的数据应用或绑定到多个元素视觉项目的用户手势的示例。

[0223] 在一个实施例中，用户交互机制 2440 准许用户按每个数据组将来自数据视觉项目 2411 的数据 2412 应用到元素视觉项目 2420 的输入参数，以便：1) 各个数据组中的一种数据组应用于第二组视觉项目中的一个不同的视觉项目的一个或多个输入参数的组，以及

2) 来自多个数据组中的每一个数据组的相同的类型单个字段被应用于对于元素视觉项目的不同的视觉项目的所述一个或多个输入参数的组中的一个输入参数。

[0224] 作为示例,参考图 24,用户可以使用单组手势来同时将数据视觉项目 2411 的数据字段 2413Aa 应用或绑定到元素视觉项目 2421 的输入参数 2421b,将数据视觉项目 2411 的数据字段 2413Ba 应用或绑定到元素视觉项目 2422 的输入参数 2422b,并将数据视觉项目 2411 的数据字段 2413Ca 应用或绑定到元素视觉项目 2423 的输入参数 2423b。如果有第四元素视觉项目,则数据字段 2413Da 可以被应用于第四元素视觉项目的相对应的输入参数。

[0225] 作为此同一组手势的结果,或或许响应于附加的手势组,可以进行进一步的应用或绑定。例如,响应于用户手势,数据视觉项目 2411 的数据字段 2413Ab 可以被应用或绑定到元素视觉项目 2421 的输入参数 2421a,数据视觉项目 2411 的数据字段 2413Bb 可以被应用或绑定到元素视觉项目 2422 的输入参数 2422a,并且数据视觉项目 2411 的数据字段 2413Cb 可以应用或绑定到元素视觉项目 2423 的输入参数 2423a。另外,数据视觉项目 2411 的数据字段 2413Ac 可以被应用或绑定到元素视觉项目 2421 的输入参数 2421c,数据视觉项目 2411 的数据字段 2413Bc 可以被应用或绑定到元素视觉项目 2422 的输入参数 2422c,并且数据视觉项目 2411 的数据字段 2413Cc 可以应用或绑定到元素视觉项目 2423 的输入参数 2423c。

[0226] 用户界面 2400 还潜在地包括具有相关联的属性 2431 的第三视觉项目 2430。第二用户交互机制 2450 准许用户使用一组手势来将第二组视觉项目 2420 合并到第三视觉项目 2430 中,以便:1) 第二组 2420 视觉项目中的每一个的一个或多个输入参数是使用第三视觉项目 2430 的相关联的属性 2431 设置的、和 / 或 2) 第三视觉项目 2430 的属性 2431 被用来改变第二组 2420 视觉项目中的每一个的一个或多个输入参数。用来实现此的手势可以是拖放操作、悬停操作,或任何其他用户手势,并甚至可以使用,完全地或部分地,被用来将来自数据视觉项目 2410 的数据应用到元素视觉项目 2420 的相同用户手势来实现。

[0227] 例如,如果被用来合并第二组 2420 视觉项目和第三视觉项目 2430 的用户手势(即,第二组手势)至少部分地与被用来将来自数据视觉项目 2411 的数据应用或绑定到元素视觉项目 2420 的输入参数的用户手势(即,第一组手势)重叠,那么,在第一组手势和第二组手势两者内有至少一个共同的手势。然而,这根本不是必需的。事实上,在第一组手势和第二组手势之间可以有没有共同的手势。如此,与将元素视觉项目 2420 与视觉项目 2430 合并相比,可以使用不同的用户动作来将来自数据视觉项目 2411 的数据应用到元素视觉项目 2420。

[0228] 在一个实施例中,每一个数据视觉项目 2411 都可以类似于在图 5 中使用图 5 的视图构造模块 521 到 524 构建的视觉项目。在该情况下,相关联的数据 2412 可以是提供到相对应的视图组件的输入参数的数据。此相关联的数据 2412 甚至可以作为视觉项目本身中的视觉被表面化,允许用户具有相关联的数据的视图。视觉项目 2430 还可类似于使用视图构造模块 521 到 524 构建的视觉项目。在该情况下,或许视觉项目 2430 的属性 2431 中的一个或多个可以使用相对应的视图组件的输入参数来设置。

[0229] 将数据视觉项目 2411 的相关联的数据 2412 应用到元素视觉项目 2420 的输入参数可以导致分析模型被解析。例如,图 4 描述了管道 200 的分析法部分 400。通过应用相关联的数据 2412,可以有需要重新计算的数据以便进一步重新填充元素视觉项目 2420 的

输入参数。另外,将元素视觉项目 2420 与第三视觉项目 2430 合并还可导致对分析法部分 400 的重新求解,以由此改变视觉项目 2420 或 2430 的输入参数。

[0230] 当然,图 24 是用户界面的抽象表示形式。现在将参考图 25 到 29 呈现准许用户指定要应用于元素视觉项目的数据视觉项目的相关联的数据并且允许元素视觉项目与另一个较高级别的视觉项目合并的用户界面的更具体的示例。

[0231] 图 25 示出了其中呈现了螺旋 2530 的用户界面 2500。螺旋 2530 是第三视觉项目 2430 的具体示例。在此情况下,螺旋 2530 的属性(作为属性 2431 的示例)可以是曲率半径,间距(或上升角)、颜色、线划宽度、线横断面图、绕组长度、开始角等等。每一个属性都可以是,例如,被提供到螺旋视图合成元素的输入属性,该螺旋视图合成元素具有构造逻辑,当执行该构造逻辑时,呈现螺旋。螺旋形状类型是可以被拖放到用户界面的工作面 2501 的多种形状中的一种。在被拖放到工作面之后,螺旋的输入参数被填充了默认值,但是,也可以改变这些默认值。

[0232] 用户可能已经将一个单独的立方体对象 2521 拖到工作面上。立方体对象 2521 是图 24 的元素视觉对象 2421 的示例。立方体对象 2521 通常具有六个矩形形状的彼此平行或垂直的面。在立方体对象 2521 被拖到工作面的除螺旋 2530 以外的另一部分中的情况下,立方体对象 2521 可能已经保留了这些基本立方体特征。然而,在此情况下,用户作出了立方体对象 2521 和螺旋 2530 将被合并的手势(或许通过使用指针 2550 将立方体 2521 拖动到螺旋 2530 的一部分上)。在此情况下,立方体 2521 的输入参数被调整,以便其中线遵循螺旋。这还可是用于定义将与螺旋视觉项目合并的元素视觉项目的类型(例如,在此情况下,立方体)的机制。

[0233] 给定了此合并操作,图 26 示出了用户界面的此特定示例的另一阶段 2600。这里,用户已经获取了数据视觉项目 2610,在此情况下,或许电子表格文档。电子表格文档 2610 是图 24 的数据视觉项目 2411 的示例。此电子表格文档包含列出了各种精神病药物的表格。数据完全是虚构的,但是提供它是为了示出此示例。第四列 2614 列出了药物的名称,每一行一个,虽然在此情况下,字段是空的,因为数据是虚构的。第三列 2613 列出了对应于每一行的药物的类别。第一列 2611 列出了对应于每一行的药物被批准用作处方的开始日期(以年表示)。第二列 2612 列出了对应于每一行的药物被批准供使用的时长(以年表示)。开始日期、时长、类别和名称是图 24 的相关联的数据 2412 的数据组的字段的示例。电子表格 2610 中的行是图 24 的相关联的数据 2412 的数据组的示例。

[0234] 这里,输入参数表 2620 看来似乎向用户示出了什么输入参数已经被选择为用于填充的目标。用户已经选择了第一列 2611(即,“开始日期”字段)以填充“螺旋上的位置”输入参数。为每一个数据组(图表 2610 中的每一行)生成立方体视觉项目。预览 2630 示出了螺旋 2530,但是,带有多个立方体元素和螺旋的合并的版本看起来像什么的预览。

[0235] 图 27 示出了用户界面 2700。这里,用户选择了各种立方体将不被裁剪,但是,将层叠在另一个立方体的顶部,以螺旋(现在不可见)充当堆的基。层叠的立方体 2710 表示元素视觉项目的示例。请注意,螺旋视觉项目的属性已经被每一个立方体的输入参数继承。另外,来自数据视觉项目 2610 的数据已经被每一个立方体继承。例如,立方体的层叠中的每一个立方体都使用螺旋上的和对应的药物的开始日期相对应的位置构建的。每一个立方体的长度被相对应的药物的时长继承。另外,立方体还从螺旋继承位置属性和曲度属性。

[0236] 图 28 示出了完整的视觉场景。在此用户界面 2800 中,可以根据相对应的药物的类别,将颜色指定给每一个弯曲的立方体。如此,相对应的药物的开始日期定义螺旋中的弯曲的立方体的开始位置。例如,用户可能已经使用了下列手势来将来自电子表格视觉项目 2411 的开始日期数据应用到立方体视觉项目:1) 选择电子表格的第一列,以及 2) 将列拖动到输入参数表 2620 的“螺旋上的位置”部分。此外,相对应的药物的时长还定义螺旋中的弯曲的立方体的长度。例如,用户可能已经使用了下列手势来将来自电子表格视觉项目 2411 的时长数据应用到立方体视觉项目:1) 选择电子表格的第一列,以及 2) 将列拖动到输入参数表 2320 的“长度”部分。最后,相对应的药物的类别定义螺旋中的弯曲的立方体的颜色。例如,用户可能已经使用了下列手势来将来自电子表格视觉项目 2411 的类别数据应用到立方体视觉项目:1) 选择电子表格的第一列,以及 2) 将列拖动到输入参数表 2620 的“颜色”部分。用户可以将电子表格的不同的字段拖放到输入参数字段的差异部分,以查看在给定情况下数据的什么视觉表示是最好的。

[0237] 通过使用此处所描述的这些原理,可以利用其他视觉元素和几何形状构建和构成复杂的几何形状。为了理解几何形状的合成,将首先描述四个基本概念。基本概念包括 1) 数据系列,2) 形状,3) 维度组,以及 4) 几何形状。

[0238] 首先,将描述数据系列。数据系列是数据上的包装器。参考图 6 示出和描述了数据系列对象的示例。数据系列对象不是数据本身。然而,数据系列知道如何枚举数据。例如,参考图 6,枚举模块 601 枚举对应于数据流对象的数据系列。另外,数据系列具有声明以下内容的属性:范围、量化,以及图的分辨率。可以被数据系列包装的数据类型的示例包括表列,表中的重复的行、层次结构、维度层次结构等等。

[0239] 形状可以从规范视图合成构建的规范视觉项目。这样的规范视觉项目的示例包括,例如,点、柱线、棱镜、气泡、表面片、图像文件、2 维或 3 维形状等等。然而,形状可以是来自数据的规范构造。来自数据的这样的规范构造的示例包括标签。可另选地,形状可以是利用数据系列和形状来填充几何形状(下面将进一步描述术语“几何形状”)的结果。

[0240] 规范形状携带潜在地允许几何形状的“绑定器-安排器”(下面所描述的)被参数化以处理多个形状的元数据。元数据还提供下面所描述的“布局助手”的暗示。元数据表示形状的各方面,诸如,形状是直线的还是曲线的,形状是平面的还是占据体积的,形状是否相对于某个维度是对称的,形状是否需要以文本意义(例如,标签)形状来阅读,是否可以利用纹理着色或叠加形状,等等。

[0241] 请注意,形状没有绝对维度,但是,可以可任选地指定其维度中的比例。例如,可以指定棱镜以具有基 L 和 W,它们是其高度 H 的某个最小百分比。形状可以可任选地指定对于如何展示形状的多个实例的约束,诸如,例如,两个柱线之间的最小距离。

[0242] 关于维度组,这与坐标系统不同。有多少要被可视化的数据的维度,维度集就包含多少维度。维度可以多于普通的欧几里得轴 x、y 和 Z,以及时间 t。在维度集中,某些可以是欧几里得的(例如,笛卡儿或极 x、y、z,或带有高度 z 的地图坐标)。稍后,当供几何形状使用时,这些欧几里得轴的效果将是描绘和控制形状实例系列内的形状如何被插入或变换为某些坐标,带有适当的比例、变换,以及分布。通过诸如聚集和层叠之类的布局效果来表明其他维度。其他维度可以涉及诸如动画、或运动速度之类的较高视觉维度。

[0243] 作为示例,考虑包括现有的国家的视觉项目。视觉项目可以各自都对应于形状

(描述国家的形状),该国家的中心的极坐标。还可有将根据国家的颜色呈现的数据系列。例如,蓝色可以表示在该国家的对外援助中花费的少量资源。红色可以表示在该国家的对外援助中花费的较大的资源。

[0244] 再一个维度可以被动画化。例如,可以有与每一个国家相关联的陀螺。可以利用关于每一个国家的政治稳定性来填充该国家的陀螺维度,不管该政治稳定性分数是如何导出的。带有较小政治稳定性的国家的陀螺被动画化,带有较慢的运动和更大的不稳定性。带有较大政治稳定性的国家的陀螺被动画化,带有更快的运动和较小不稳定性。其他非欧几里得维度可包括纹理。

[0245] 维度集可包括可见性 / 条件性隐藏、可滚动性、标题、标签、细分、可绘制的粒度,准许的范围(例如,是否允许负的)等等的声明。

[0246] “几何形状”包括容器和一个或多个绑定器 - 安排器。

[0247] 至于容器,几何形状包含其中数据将通过形状被可视化的容器的视觉元素 / 布置的描述。例如,最简单的可能的柱状图指定包围矩形的形状、容器的 2/3D 维度中的比例(可另选地,绝对维度),坐标系统(例如,笛卡儿)。更复杂的可视化可以指定附加的概念。例如,地图可以指定几何形状内的限制可以将形状放在坐标系统内的什么位置的细分(如区域、街道)。象限图可以指定其笛卡儿坐标轴适应负值、颜色、纹理、透明度,及其他可控制的视觉参数。

[0248] 容器可以可任选地指定潜在地允许相对应的绑定器 - 安排器中所需的智能的有效部分被析出到布置助手的小集合(常常基于解算器)的元数据。元数据表示容器的各方面,诸如,例如:欧几里得轴(有更多轴类型)是直线的还是曲线的,容器是平面的还是占据体积的,容器是否相对于某个维度是对称的,等等。

[0249] 其次,几何形状携带“绑定器 - 安排器”的声明,该“绑定器 - 安排器”知道如何:1) 通过将传入的数据系列和原始形状应用到 DataShape BindingParams 来生成形状实例系列,该 DataShapeBindingParam 描述了数据值如何映射到形状的维度(例如,高度)或视觉属性(例如,颜色),和 / 或基于数据值来如何从一组形状中进行选择,以及 2) 将传入的轴集映射到容器的坐标系统,并映射到容器的其他视觉元素(层叠、聚集、着色、运动等等), 3) 按每个 ShapeAxis BindingParam 将形状实例系列布置到一个或多个维度上,如被映射到容器,以及 4) 在必要时在展示的形状之间内插,例如,连接线,或从小的 surface-lets 或 patches 创建连续的表面。

[0250] 被填充的几何形状(即,利用一个或多个数据系列和相对应的形状实例化的几何形状)本身可以被视为形状,以便被传递到另一个几何形状中。例如,在层叠的 / 聚集的柱状图中:最里边的几何形状具有简单容器,可以在其中层叠其他柱线的柱线。填充最里边的几何形状会导致进入第二几何形状的形状,其中,容器是柱线的簇。最后,由对第二几何形状的填充所产生的形状被馈送到最外面的几何形状,该几何形状知道如何沿着水平轴布置形状,还示出它们在 y 轴上的高度。

[0251] 在上面的层叠的 / 聚集的柱状图示例中,最里边的几何形状只具有高度,第二几何形状具有(排序或未排序的)聚集,而最外面的几何形状具有简单(即,未层叠的)垂直维度和简单(即,没有察觉到聚集)水平维度。然而,合成层叠的 / 聚集的柱状图也可以被视为单个几何形状,该单个几何形状可以包括带有三个方面的维度集:层叠感知的高度维

度、聚集维度,以及水平 X 维度。

[0252] 布局助手依赖于算法或解算器,并帮助确定形状在包含性的几何形状内的“正确的”或“最佳的”定位,以及形状之间的内插。如上文所提及的,布局助手的思想是降低绑定器-安排器内的特殊性。有两种类型的布局助手,在特定绑定器-安排器的级别调用的本地布局助手,以及检查整个包含性的几何形状的全局布局助手(包括在将形状放入在几何形状内的多个 B-A 的中间结果处)。

[0253] 如此,管道允许跨各种领域的复杂交互。

[0254] 附加示例应用程序

[0255] 图 1 和 2 的体系结构可以允许构建无数的数据驱动的分析法模型,而不管领域如何。这些领域根本无需类似。无论哪里有问题需要求解,只要对视觉项目应用分析法是有帮助的,则此处所描述的原理就可以是有益的。直到现在,只描述了几个示例应用程序,包括风水房间布局应用程序。为说明此处所描述的原理的广泛的适用性,现在将描述几个附加的广泛的并且不同的示例应用。

[0256] 附加示例 #1——零售货架安排

[0257] 产品销售人员常常使用 3-D 可视化来销售货架上的商品,最终显示和新促销商品。利用管道 201,销售人员将能够在现场进行“假设分析”。给定某些产品布局并给定最小日销售额/纵尺(linear foot)阈值,销售人员可以计算和可视化手边的最小所需库存。相反,给定手边的某个库存并给定双周补充周期,销售人员可以计算将给出所需销售额/纵尺的产品布局。零售商将能够使这一影响可视化,比较各个情形,并比较利润。图 29 示出了示例零售商货架安排可视化。输入数据可包括产品的视觉图像、产品数量、为每一个产品分配的线性平方尺,和每一个产品的货架号等等。图 29 的示例示出了虚拟世界内的图表的该应用。这里,平面图 2901 和架子布局 2902 可以在分析上是相关联的,以便架子布局 2902 中的变化影响平面图 2901,反之亦然。

[0258] 附加示例 #2——城市规划

[0259] 城市规划混搭(mash up)变得很重要。使用此处所描述的原理,可以将分析法集成到这样的解决方案中。城市规划者将打开由专家所创建的交通模型,并从道路改善长廊(gallery)拖进一个桥梁。该桥梁将带来分析行为,诸如长度约束和强风操作限制。通过适当的可视化,规划者将看到并比较不同的桥梁类型和布局对交通量的影响。可以将此处所描述的原理应用于其中地图可以用于各种用途的任何地图情形。地图可以用于理解地形的特征并查找到某个位置的方向。地图也可以是用于比较分地区的数据的视觉背景幕。最近,地图被用于创建虚拟世界,其中可以在地图中覆盖或放置建筑物、室内装饰和任意的 2-D 或 3-D 对象。图 30 示出了示例可视化城市规划图。请注意,图表再次可以在功能上和分析上与虚拟世界(在此情况下,城市)的地图集成。随着图表变化,虚拟世界也会变化,反之亦然。

[0260] 附加示例 #3——视觉教育

[0261] 在其中复杂数据不仅需要由领域从业者理解而且还应由公众理解的诸如科学、医学、以及人口统计之类的领域,作者可以使用此处所描述的原理来创建激起受众的兴趣的数据可视化。他们将使用领域特定的比喻,并提供作者的风格。图 31 是关于儿童的教育插图。图 32 是关于人口密度的常规插图。通常,这样的可视化仅仅是静态插图。利用此处

所描述的原理,这些可以变成实况的交互式体验。例如,通过输入在地理上分布的增长模式来作为输入数据,用户可以看到人口峰值变化。创作的模型支持的某些可视化将让用户执行“假设分析”。即,作者可以改变某些值,并看到该变化对其他值的影响。

[0262] 附加示例 #4——将视图组件应用到参数目标

[0263] 在某些情况下,可能需要将视图组件应用到各种参数目标(例如,其他视图组件),例如,如图 33 和 34 所示。具体而言,图 33 和 34 两者中的面板的高度表示拿破仑在 1812 年的不幸的在俄国的战役中的军队规模。在图 33 中,将面板应用于表示拿破仑的军队采取的实际路线的参数目标。在图 34 中,将面板应用于一个不同的参数目标:螺旋。如此,可以将视图组件(如面板)应用于不同的参数目标(如,表示拿破仑的军队采取的实际路线的视图组件或表示螺旋的视图组件)。

[0264] 视图组件可以是它们被应用到的参数目标的子。例如,图 33 中的面板可以是军队路线的子,而图 34 中的面板可以是图 34 中的螺旋的子。另外,表示军队规模的标签可以是其高度表示该军队规模的面板的子。

[0265] 理想情况下,与子视图组件的属性关联的解算器和与父视图组件的属性关联的解算器可以通过依赖树显式地编写,或通过包括这样的解算器的属性-设置器的相互关系隐式地编写。

[0266] 如此,设置子中的属性可以触发对父中的属性的重新求解(有时被称为“提高”),而设置父中的属性可以触发对子中的属性的重新求解(有时被称为“委派”)。例如,参考图 33,用户可以尝试改变面板的高度,这可以触发面板的属性-设置器以增大面板的比例。另外,面板的比例属性-设置器可以调用面板组的比例的属性-设置器(提高)。面板组的比例属性-设置器可以为每一个面板调用单个比例属性-设置器(委派)。

[0267] 请注意,具体而言,图 33 是将图表应用到虚拟世界的示例。“虚拟世界”是假想的地图的某些实物的计算机化表示。地图可以是二维的,或者也可以是三维的。例如,城市地图可包括以地理关系展示的街道、建筑物、河流等等。另一方面,“图表”表示应用于变量的数据系列。可以通过将如图表所示的数据系列应用到虚拟世界的某些方面,使用此处所描述的原理将图表应用于虚拟世界。该虚拟世界方面可以是水平或垂直面,三维对象的外部服务、路线等等。虚拟世界可以表示实际的物理空间,或者也可以只表示假设的区域。虚拟世界可以表示城镇、城市、州、省、地区、建筑物、邻近地区、行星中的全部或一部分,或任何其他物理区域。

[0268] 例如,在图 33 的示例中,图表数据包括,对于许多周期中的每一个,在该时间拿破仑的军队中的人数。此信息可以适用于拿破仑的军队的路线。每一个柱线的高度都表示拿破仑的军队在军队的特定旅程段时的士兵数量。在需要时,在地图中可以包括其他地理特征,如河流、城市、海洋等等。通过将诸如坐标系统、轴或标志器之类的图表特征应用到诸如表面之类的虚拟世界特征,可以将图表应用于虚拟世界。

[0269] 作为图表应用到虚拟世界的其他示例,假设一个公司具有表示雇员情绪的数据。此图表数据可以适用于虚拟世界,如通过呈现每一个雇员的窗口的颜色,公司场地的三维呈现,某种颜色代表一种情绪。例如,蓝色窗口可以表示快乐的雇员,而红色窗口可以表示不快乐的雇员。可以进行适当的分析。例如,可以取场地地图的高级视图,并确定哪些建筑物趋向于红色。或许一个建筑物与其他建筑物隔离,导致较少的交互机会,以及较低的情

绪。公司可以重新考虑是否在将来租用该建筑物,而试图为那些雇员获得和建筑物总体更为接近的建筑物。

[0270] 因此,此处所描述的原理提供了可视化的问题求解和分析领域的主要范式转变。该范式转变适用于所有领域,因为此处所描述的原理可以适用于任何领域。

[0271] 对数据的领域特定的分类

[0272] 回顾图 2,管道 201 是数据驱动的。例如,输入数据 211 被提供到数据部分 210,分析法数据 221 被提供到分析法部分 220,而视图数据 231 被提供到视图部分 230。描述了这些数据中的每一个的示例。只要说可以由创作组件 240 选择的数据量可以十分大就够了,特别是假设其中模型的某些部分可以被导入到模型中以构成越来越多复杂的模型的合成比较容易。为有助于在数据中导航,以便可以选择适当的数据 211、221 和 231,分类组件 260 提供输入数据的许多领域特定的分类。

[0273] 图 35 示出了分类组件 260 可以在其中操作的分类环境 3500。分类涉及将项目分类为类别并将这些类别相关联。如此,环境 3500 包括将接受分类的项 3510 的集合。在图 35 中,项 3510 的集合被示为总共只包括几个项,包括项 3511A 到 3511P(统称为“成员项 3511”)。虽然只示出了几个成员项 3511,但是,可以有应该被分类的任何数量的项,或许甚至数百、数千或者甚至数百万项,如省略号 3511Q 所示。成员项 3511 包括成员项的池,从该池中,创作组件 240 可以选择以便向管道 201 提供数据 211、221 和 231。

[0274] 领域敏感的分类组件 3520 访问成员项 3511 的全部或一部分,还能够生成成员项 3511 的不同的分类。例如,分类组件 3520 生成领域特定的分类 3521。在此情况下,有五个领域特定的分类 3521A 到 3521E,潜在地还有如省略号 3521F 所示的其他属性。还可少于五个的由分类组件 3520 创建和管理的领域特定的分类。

[0275] 作为示例,分类 3521A 可以分类适于“风水”领域的成员项,分类 3521B 可以分类适于摩托车设计领域的成员项,分类 3521C 可以同样适用于城市规划领域,分类 3521D 可以适于库存管理领域,而分类 3521E 可以适于摘要插图领域。当然,这些只是可以由管道 201 服务的潜在地无数的领域中的五个。分类中的每一个都可以使用可用的成员项的全部或子集来在相对应的分类中进行分类。

[0276] 图 36 示出了成员项目的分类的一个具体而简单的示例 3600。例如,分类可以是图 35 的领域特定的分类 3521A。随后的图将阐述更复杂的示例。分类 3600 包括类别节点 3610,该类别节点 3610 包括成员项 3511 的除成员项 3511A 和 3511E 之外的全部。类别节点 3610 可以是例如包括指向组成成员项的指针的对象,如此,在逻辑意义上,成员项可以被视为“包括在类别节点 3610 内”。类别节点 3610 还具有与其相关联的属性关联描述符 3611,该描述符 3611 使用候选成员项的属性来描述类别节点 3610 的成员身份资格。当确定是否应该在类别中包括一个成员项时,可以使用属性关联描述符来针对该成员项的属性评估描述符。

[0277] 在一个分类中,可以以许多不同的方式将两个类别彼此相关联。一个常见的关系是,一个类别是另一个类别的子集。例如,如果有包含表示交通工具的所有对象的“交通工具”类别,则可以有包含交通工具类别的子集的“小汽车”类别。两个类别的属性关联描述符可以定义特定关系。例如,交通工具类别的属性关联描述符可以指示具有下列属性的对象将被包括在类别中:1) 对象是可移动的,2) 对象可以包含人。小汽车类别属性关联描述

符可要么明确地要么隐式地包括这两个属性要求,还可包括下列属性要求:1) 对象包含至少3个在对象的运动过程中与地球保持接触的轮子,2) 对象是汽车,3) 对象的高度不超过6英尺。基于每一个类别的属性关联描述符,分类组件可以将一个对象分配到任何给定领域特定的分类中的一个或多个类别,还可理解类别之间的关系。

[0278] 在图 36 中,示出了包括另一个属性关联描述符 3621 的第二类别节点 3620。类别节点 3620 在逻辑上包括满足属性关联描述符 3621 的所有成员项。在此情况下,类别节点中在逻辑上所包括的成员项是第一类别节点 3610 中所包括的成员项的子集(例如,包括成员项 3511F、3511J、3511N 和 3511P)。这可能是因为,第二类别节点 3620 的属性关联描述符 3621 指定与第一类别节点 3610 的关联描述符 3611 相同的属性要求,除一个或多个附加属性要求之外。第一类别节点 3610 和第二类别节点 3620 之间的关系通过关系 3615 在逻辑上来表示。

[0279] 在交通工具-小汽车示例中,类别之间的关系是子集关系。即,一个类别(例如,小汽车类别)是另一个类别(例如,交通工具类别)的子集。然而,还有各种其他类型的关系,甚至或许之前从来没有被识别或使用的新关系。例如,可以有大多数继承关系,其中,如果一个类别中的大多数(或某个指定百分比的)对象具有特定属性值,则另一个类别中的对象具有此属性,并继承此属性值。可以有“类似的颜色”关系,其中,如果一个类别对象具有某一波长范围的可见光内的基色,那么,另一个类别包含具有某一相邻波长范围的可见光内的基色的对象。可以有“病毒突变”关系,其中,如果一个类别包含表示主要由一种特定病毒所导致的某些传染病的对象,相关联的类别可包括表示由该病毒的变异形式所导致的某些传染病的对象。示例可以有很多。本领域技术人员将认识到,在阅读了本描述之后,类别之间的关系的类型是不受限制的。

[0280] 此外,一个分类可以具有许多不同类型的关系。为清楚起见,现在将以摘要方式描述各种分类。在图 37A 到 37C 中示出了抽象地表示的分类的示例。然后,将描述具体示例,理解此处所描述的原理允许领域特定的分类在数据驱动的可视化中的无数的应用。

[0281] 图 36 的示例是简单的两个类别节点分类,而图 37A 到 37C 的示例更复杂。图 37A 到 37C 的分类 3700A 到 3700C 中的每一个节点都表示包含零个或更多个成员项的类别节点,并可以具有与每一个相关联的属性关联描述符,该描述符基本上是由于允许成员项进入类别节点的许可策略。然而,为避免不适当的复杂性,没有示出分类 3700A 到 3700C 的类别节点中的每一个的成员项和属性关联描述符。类别节点之间的线表示类别节点之间的关系。它们可以是子集关系或某种其他类型的关系,而无限制。类别节点之间的关系的准确特征不是关键的。尽管如此,为强调在分类中的类别节点之间可以有各种关系类型,关系被标记有 A、B、C、D 或 E。

[0282] 图 37A 到 37C 只作为示例来提供。图 37A 到 37C 的分类的准确的结构不仅是不是关键,而且此处所描述的原理准许甚至基于相同输入候选成员项的组来生成什么样的分类的比较大的灵活性。在这些示例中,分类 3700A 包括使用关系类型 A、B 和 C 彼此相关联的类别节点 3701A 到 3710A。分类 3700B 包括使用关系类型 B、C 和 D 彼此相关联的类别 3701B 到 3708B。分类 3700C 包括使用关系类型 C、D 和 E 彼此相关联的类别 3701C 到 3712C。在此示例中,分类 3700A 和 3700B 是分层的,而分类 3700C 在更大程度上是非分层网络。

[0283] 随着新的候选成员项变得可用,可以针对每一分类中的每一个类别节点的属性关

联描述符,评估那些候选成员项。如果成员项的属性具有准许成员项满足属性关联描述符的要求(即,许可策略)的值,则成员项被吸收入类别节点。例如,或许指向成员项的指针被添加到类别节点中。如此,如果新成员项具有足够数量的属性,则可以将新成员项自动导入到所有分类中的适当的类别中。

[0284] 图 38 示出了包括多个属性 3801 的成员项目 3801。可以有单个属性,但是,还可有潜在地数千与成员项 3800 相关联的属性。在图 38 中,成员项 3800 被示为包括四个属性 3801A、3801B、3801C 和 3801D,潜在地还有如省略号 3801F 所示的其他属性。对于这些属性可以是什么没有限制。它们可以是对于将成员项分类为一个分类有用的任何东西。

[0285] 在一个实施例中,可以分类数据部分 210、分析法部分 220,以及视图部分 230 中的每一个的潜在的数据。例如,考虑其中作者正在编写允许个人(如消费者或街坊)与城市的地图连接的消费者应用程序的领域。

[0286] 在此消费者领域中,可以有可以被选择的视图数据 231 的分类。例如,可以有包括全部建筑物的建筑物类别。可以有不同类型的建筑物:政府大楼、医院、餐厅、住宅等等。还可有包括铁路、道路,以及运河子类别的运输类别。道路类别可以包含表示街道、高速公路、自行车道、天桥等等的类别或对象。街道类别可包括单向街道、多行街道、转弯车道、中央车道等等的视觉表示的对象或类别。可以有停车场类别,示出了不同类型的停车场的视觉表示或停车场的其他子类别(例如,多层停车场、地下停车场、街道停车场、停车场等等)。还可根据停车场是免费的还是收费的来对停车场进行子分类。

[0287] 在此消费者领域中还可有输入数据的分类。例如,多层停车场可以具有与它相关联的数据,诸如,例如,1) 停车场是否有代客泊车,2) 停车场每小时的费用是多少,3) 停车场开放的小时数,4) 停车场是否有安全巡逻,如果有,停车场每一单位面积有多少保安员,5) 停车场的层数,6) 如果只有一层,则停车场的平方尺,如果是多层停车场,则每一层的平方尺,7) 多层停车场发生的按年计的历史汽车失窃数量,8) 停车场的用量,9) 停车是否只限于满足一个或多个条件(即,在附近的企业使用,光顾餐厅或商场等等),或可以有用的任何其他数据。还可有与其他视觉项目相关联的数据,以及可能决不会影响呈现视觉项目的方式,但是可以在某一时刻用于计算的数据。

[0288] 然而,还可有对此消费者地图领域特定的分析法数据 221 的分类。例如,分析法可以呈现一个类别中的基于成本的分析法、另一个类别中的基于时间的分析法、再一个类别中的基于距离的分析法、另一个类别中的目录分析法,以及另一个类别中的路线分析法。这里,分析法被分类以帮助作者为所希望的应用形成分析模型。例如,路线分析法类别可包括计算路线的公式的类别,指定可以对路线作出什么限制的约束(如最短路线,高速公路的最多使用、回避街道等等),或规则(如特定道路上的交通方向)。对于其他类别,也可包括类似的子类别。

[0289] 现在考虑另一个领域,也处理城市的布局,但是,此时,领域是城市规划。这里,有城市规划者感兴趣的而消费者很少感兴趣或不感兴趣的分析法。例如,可能有这样的分析法:该分析法计算在给定某一交通流量的情况下人行道应该有多宽,被置于某一面积中的某一道路类型的每个纵尺的总体安装和维护成本是什么,在给定未来 20 年的预期的交通流量模式预测的情况下,桥的安全因素是什么,在当前城市规划中交通流量瓶颈是什么,如果在特定位置构建特定建筑物,环境影响将是什么样的,如果对该特定建筑物的使用施加

某些限制,将有什么影响等等。这里,要被求解的问题不同于在消费者领域要被求解的那些问题。因此,与消费者领域相比,对于城市规划领域,对分析法的分类可以有很大的不同,尽管两者都处理城市拓扑。

[0290] 另一方面,牵引车设计领域可能对整个不同的分析法组感兴趣,并将使用不同的分类。例如,牵引车设计领域的视觉项目可能完全不同于城市规划领域的视觉项目。不再对于城市视觉元素的关心。现在,构成牵引车的各种视觉元素被分类。作为示例,可以使用什么可以连接到什么的关系来分类视觉项目。例如,可以有“可以连接到座位的东西”,“可以连接到汽化器的东西”,“可以连接到后轴的东西”等等的类别。还可有不同的分析法分类。例如,可以有关于轮胎上的胎面花纹深度的约束,假设牵引车需要在湿土上导航。可以有计算牵引车或其子组件等等的总重量的分析法。

[0291] 图 39 示出了领域特定的分类 3900 并表示图 35 的领域特定的分类 3521 的一个示例。在一个实施例中,领域特定的分类包括其中至少一些可用的数据项被分类为相对应的相关联的类别的数据分类 3901,其中至少一些可用的视图组件被分类为相对应的相关联的视图组件类别的视图组件分类 3903,以及其中至少一些可用的分析法被分类为相对应的相关联的分析法类别的分析法分类 3902。已经描述了其中以领域特定的方式对数据、分析法以及视图组件进行分类的这样的领域特定的分类的示例。

[0292] 图 40 示出了用于导航和使用分析法的方法。与相对应的领域特定的分析法分类(动作 4003)一起访问(动作 4001)分析法组件 220。如果有多个领域特定的分析法分类,则在可以访问领域特定的分析法分类(动作 4003)之前,首先标识领域(动作 4002)。

[0293] 然后,可以通过遍历相关联的类别,导航(动作 4004)分析法分类。此导航可以由人在计算系统的协助下执行,或者也可以甚至由计算系统单独执行,而无需人的同时的协助。计算机或人可以从关联属性描述符导出要被输入到每一个类别中的分析法的许可策略的该类别的信息。信息还可由类别之间的关系导出。可以使用导航来解决分析法问题,从而求解输出模型参数,或者或许为了合并来自多个模型的分析法。可另选地,可以一开始使用导航来编写分析法模型。

[0294] 例如,假设分析法分类就要被解决的问题的类型的身份而言来分类关系。编写者可以通过查看相关的问题类型类别中的所有那些分析法来开始。该类别可以具有定义要被求解的问题的某些部分的相关联的类别。用户可以快速地导航到那些相关联的类别,并查找该领域中的相关的分析法。

[0295] 搜索和探查

[0296] 如上文所提及的,可以使用数据驱动的分析法模型来构成分析法密集搜索和探查操作。图 41 示出了用于使用数据驱动的分析法模型来进行搜索的方法 4100 的流程图。每当搜索工具 242 接收到或以其他方式访问搜索请求时,都可以执行方法 4100(动作 4101)。

[0297] 此处所描述的原理不仅限于用户可以用来输入搜索请求的机制。尽管如此,现在将提供几个示例以显示出范围广泛的搜索请求输入机制。在一个示例中,或许搜索请求是基于文本的,并被直接输入到文本搜索字段中。在另一示例中,或许单选按钮被填充以输入搜索参数。或许,也可以使用滑块来输入搜索参数的范围。搜索请求可能是以和用户交互的方式生成的。例如,在用户请求体验到某一噪音级别范围的房地产的情况下,应用程序可以生成变得越来越响的噪声,并要求用户在噪声变得比用户希望忍受的更响时按“噪音过高”

按钮。

[0298] 搜索请求不是常规的搜索请求,而是可能要求数据驱动的分析法模型的求解操作。然而,在求解之前,图 2 的搜索工具 242 标识应该被求解以便能够对请求作出响应的任何模型参数(动作 4102)。这可以使用,例如,上文所讨论的各种分类来做到。例如,在用户正在搜索在一年中的上午 9:15 之后的任何时间都不在高山的阴影里的房地产的情况下,可以有叫做“高山阴影”的被求解的模型变量。在用户搜索体验到某些噪音级别的房地产的情况下,在给定特定坐标的情况下,可以有要被求解的叫做“平均噪音”的模型变量。

[0299] 一旦标识了相关输出变量,就使用分析法部分 220 的分析关系来求解输出变量(动作 4103)。然后,搜索工具 242 使用已求解的输出变量来,使用已求解的值,表达对搜索请求的响应(动作 4104)。虽然在某些情况下用户可以在方法 4100 正在被执行过程中与方法 4100 进行交互,但是,方法 4100 可以由计算系统来执行,而无需来自人的同时协助。搜索请求可以由用户或者或许甚至由另一个计算或软件模块所发出。

[0300] 每当处理搜索请求时,方法 4100 可以重复很多次。对于每一个搜索请求,被求解的模型变量可以不同,但是不必是不同的。例如,对具有某一价格范围和噪音级别的房屋可以有三个搜索请求。例如,可以有对 \$400,000 到 \$600,000 价格范围内并且其平均噪音级别低于 50 分贝的房屋的搜索请求。这里要被求解的参数将是噪音级别。第二搜索请求可以是对 \$200,000 到 \$500,000 价格范围内并且其平均噪音级别低于 60 分贝的房屋的搜索请求。这里要被求解的参数将再一次是噪音级别。然而,请注意,在第二搜索请求中,一些求解操作对于搜索请求已经执行。例如,基于第一搜索请求,系统已经标识了 \$400,000 到 \$500,000 价格范围内并且其平均噪音级别低于 50 分贝的房屋。如此,对于那些房屋,没有必要重新计算噪音级别。一旦被求解,就可以保留那些值,用于进行未来的搜索。如此,这允许用户通过提交后继的请求来执行探查。然后,用户可以提交对 \$400,000 到 \$500,000 价格范围内并且具有小于 45 分贝的噪音级别的房屋的第三搜索请求。由于那些房屋的噪音级别已经被求解,因此,没有必要再次求解它们。因此,可以以少得多的计算返回搜索结果。本质上,系统可以通过解决问题来了解新信息,并能够利用该新信息以解决其他问题。

[0301] 如上文所提及的,每一个搜索请求都可以涉及求解不同的输出模型变量。例如,在执行刚刚所描述的搜索请求之后,用户可以提交对不在高山的阴影里的房屋的搜索请求。一旦系统求解了此,每当此用户或另一个用户提交类似的请求时,可以使用来自该求解的结果来完成该随后的搜索请求。用户可以提交对将抵抗 8.0 级地震的房屋的搜索请求,导致模拟验证每一个房屋将保持不倒、倒下,或者或许提供房屋将保持不倒的某种百分比机率。对于那些没有足够的结构信息难以执行准确的模拟的房屋,系统可以简单地声明,结果是非决定性的。一旦执行了地震模拟,除非有要求重新模拟的对房屋的某种结构改变,或者除非有某种改善的仿真解算器,每当某人提交对可以承受某一地震震级的房屋的搜索请求时,都可以使用这些结果。用户还可以执行对某一价格范围的在万一发生 5 级飓风的情况下将不能被淹没或毁坏的房屋的搜索请求。

[0302] 已经较详细地描述了各实施例,作为边注,此处所描述的各种操作和结构可以,但是不必通过计算系统来实现。因此,为结束本描述,将参考图 25 描述示例计算系统。

[0303] 图 42 示出了计算系统 4200。计算系统现在越来越多地采取各种形式。计算系统可以,例如,是手持式设备、电器、膝上型计算机、台式计算机、大型机、分布式计算系统,或

者甚至常规上不被视为计算系统的设备。在此说明书和权利要求书中,术语“计算系统”被广泛地定义为包括其中包括至少一个处理器以及能够在其上具有可以由处理器执行的计算机可执行指令的存储器的任何设备或系统(或其组合)。存储器可以采取任何形式,并可以取决于计算系统的性质和形式。计算系统可以分布在网络环境中,并可包括多个组成计算系统。

[0304] 如图 42 所示,在其最基本的配置中,计算系统 4200 通常包括至少一个处理单元 4202 和存储器 4204。存储器 4204 可以是物理系统存储器,可以是易失性的、非易失性、或两者的某种组合。此处也可以使用术语“存储器”来指诸如物理存储介质之类的非易失性大容量存储。如果计算系统是分布式的,则处理、存储器和 / 或存储能力也可以是分布式的。如此处所使用的,术语“模块”或“组件”可以是指是在计算系统上执行的软件对象或例程。此处所描述的不同组件、模块、引擎,以及服务可以实现为在计算系统上执行的对象或进程(例如,作为分离的线程)。

[0305] 在随后的描述中,参考由一个或多个计算系统执行的动作描述了各实施例。如果这样的动作是以软件实现的,则执行该动作的相关联的计算系统的一个或多个处理器响应于已经执行了计算机可执行指令,引导计算系统的操作。这样的操作的示例涉及对数据的操纵。计算机可执行指令(以及被操作的数据)可以存储在计算系统 4200 的存储器 4204 中。

[0306] 计算系统 4200 还可以包含可允许计算系统 4200 诸如通过例如网络 4210 等与其他消息处理器进行通信的通信信道 4208。通信信道 4208 是通信介质的示例。通信介质通常以诸如载波或其它传输机制等“已调制数据信号”来体现计算机可读指令、数据结构、程序模块或其它数据,并包括任何信息传送介质。作为示例而非限制,通信介质包括有线介质,如有线网络或直接线连接,以及诸如声学、无线、红外线及其他无线介质之类的无线介质。如这里所使用的术语“计算机可读介质”包括存储介质和通信介质两者。

[0307] 本发明的范围内的实施例还包括用于携带或其上储存有计算机可执行指令或数据结构的计算机可读介质。这样的计算机可读介质可以是可由通用或专用计算机访问的任何可用介质。作为示例而非限制,这样的计算机可读介质可包括诸如 RAM、ROM、EEPROM、CD-ROM 之类的物理存储器和 / 或存储介质或其他光盘存储、磁盘存储或其他磁存储设备、或可用于携带或存储计算机可执行指令或数据结构形式的所需程序代码装置且可由通用或专用计算机访问的任何其他介质。上述的组合也应被包括在计算机可读介质的范围内。

[0308] 计算机可执行指令例如包括,使通用计算机、专用计算机、或专用处理设备执行某一功能或某组功能的指令和数据。尽管用结构特征和 / 或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于此处所描述的具体特征或动作。相反,此处所描述的具体特征和动作是作为实现所要求保护的主题的示例形式来公开的。

[0309] 本发明可具体化为其它具体形式而不背离其精神或本质特征。所描述的实施例在所有方面都应被认为仅是说明性而非限制性的。从而,本发明的范围由所附权利要求书而非前述描述指示。落入权利要求书的等效方案的含义和范围内的所有改变应被权利要求书的范围所涵盖。

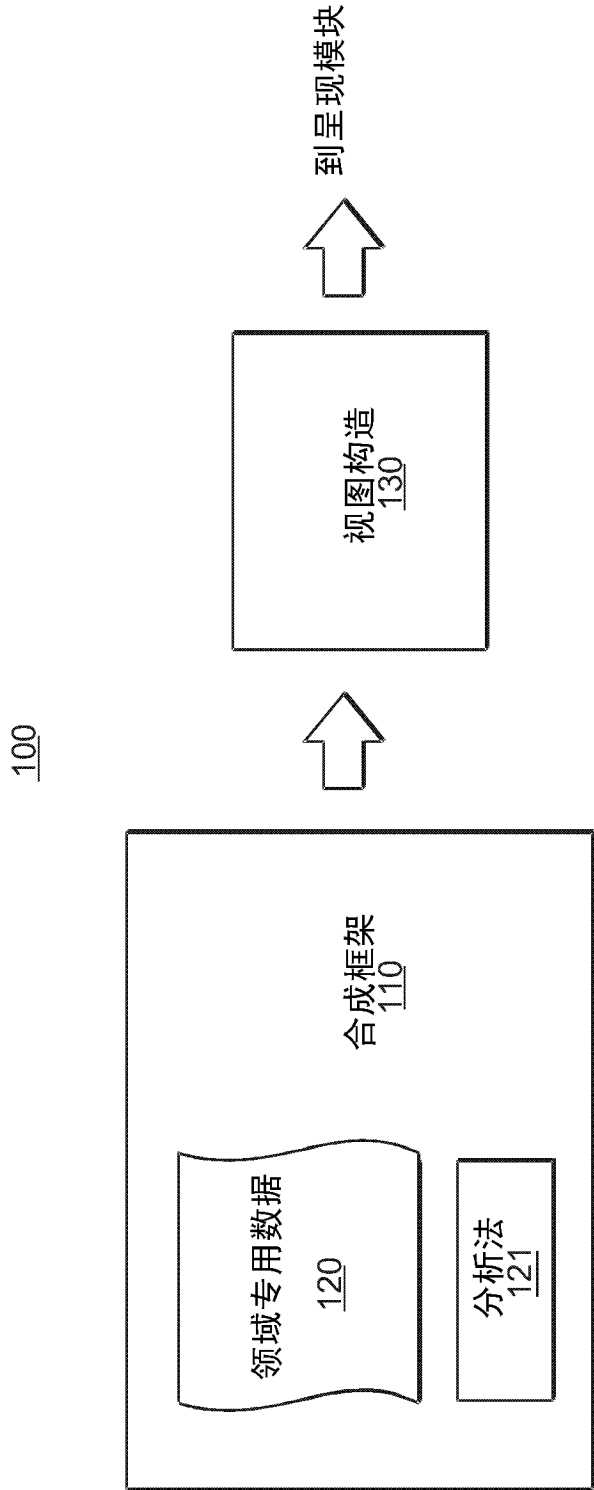


图 1

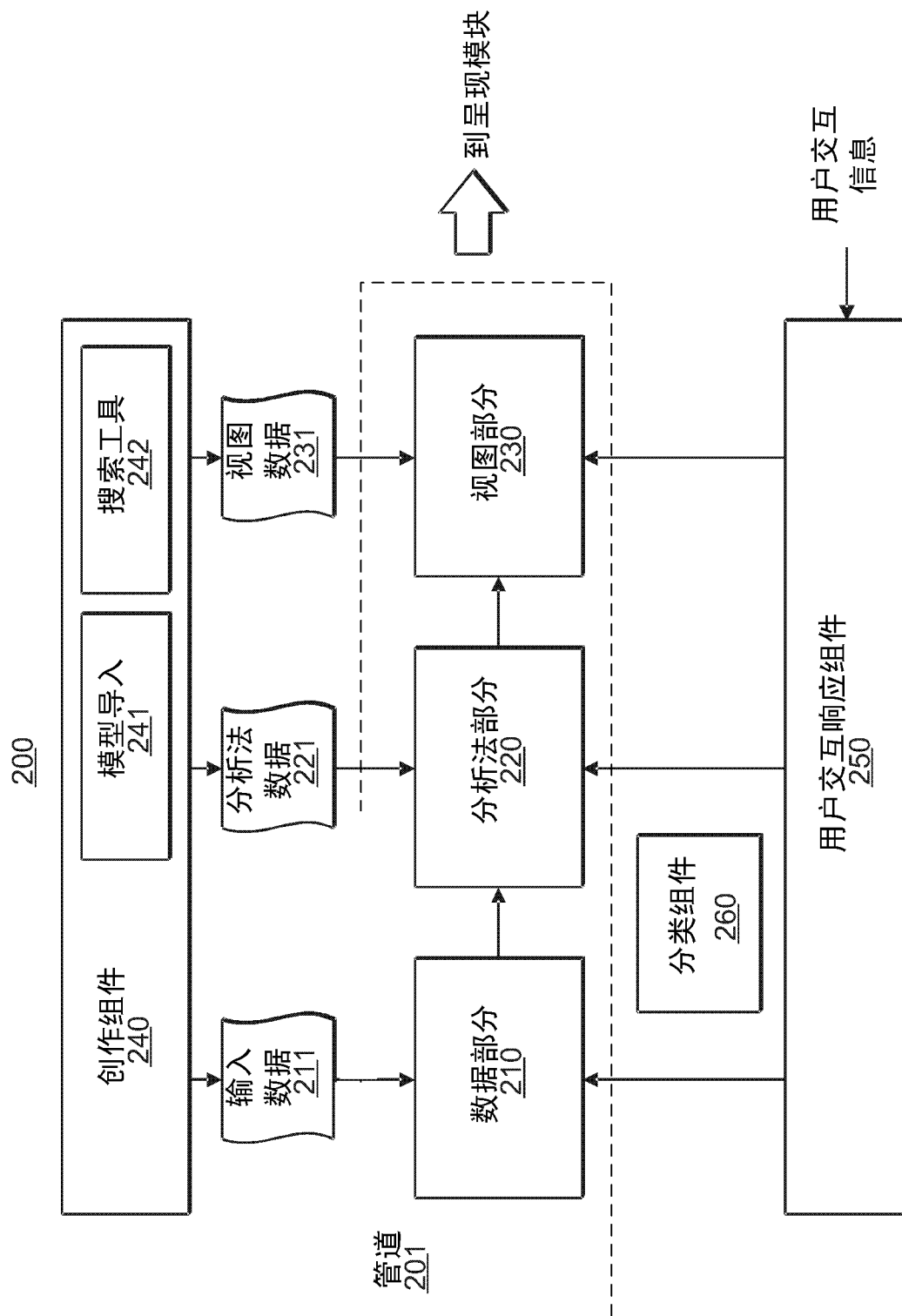


图 2

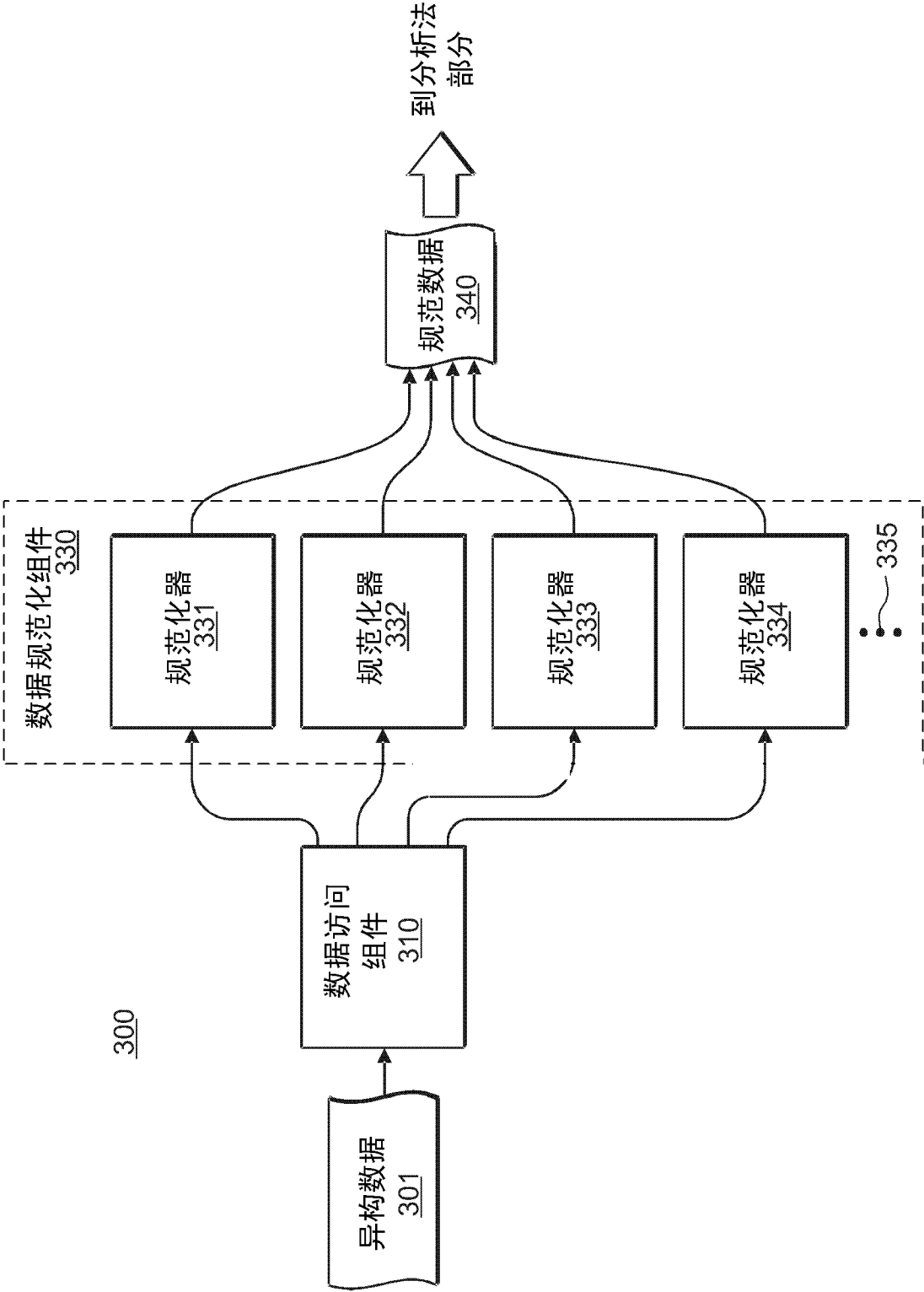


图 3

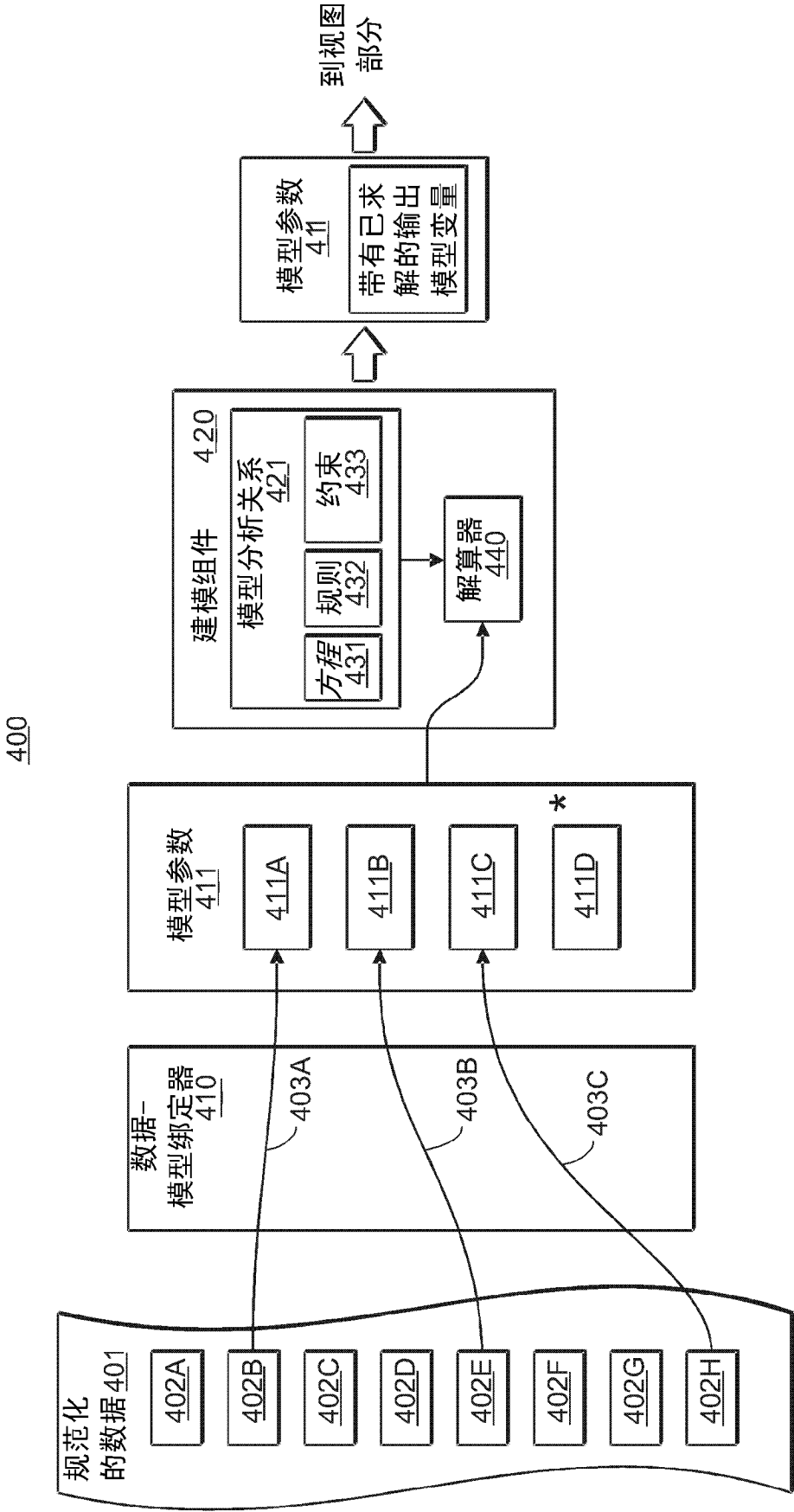


图 4

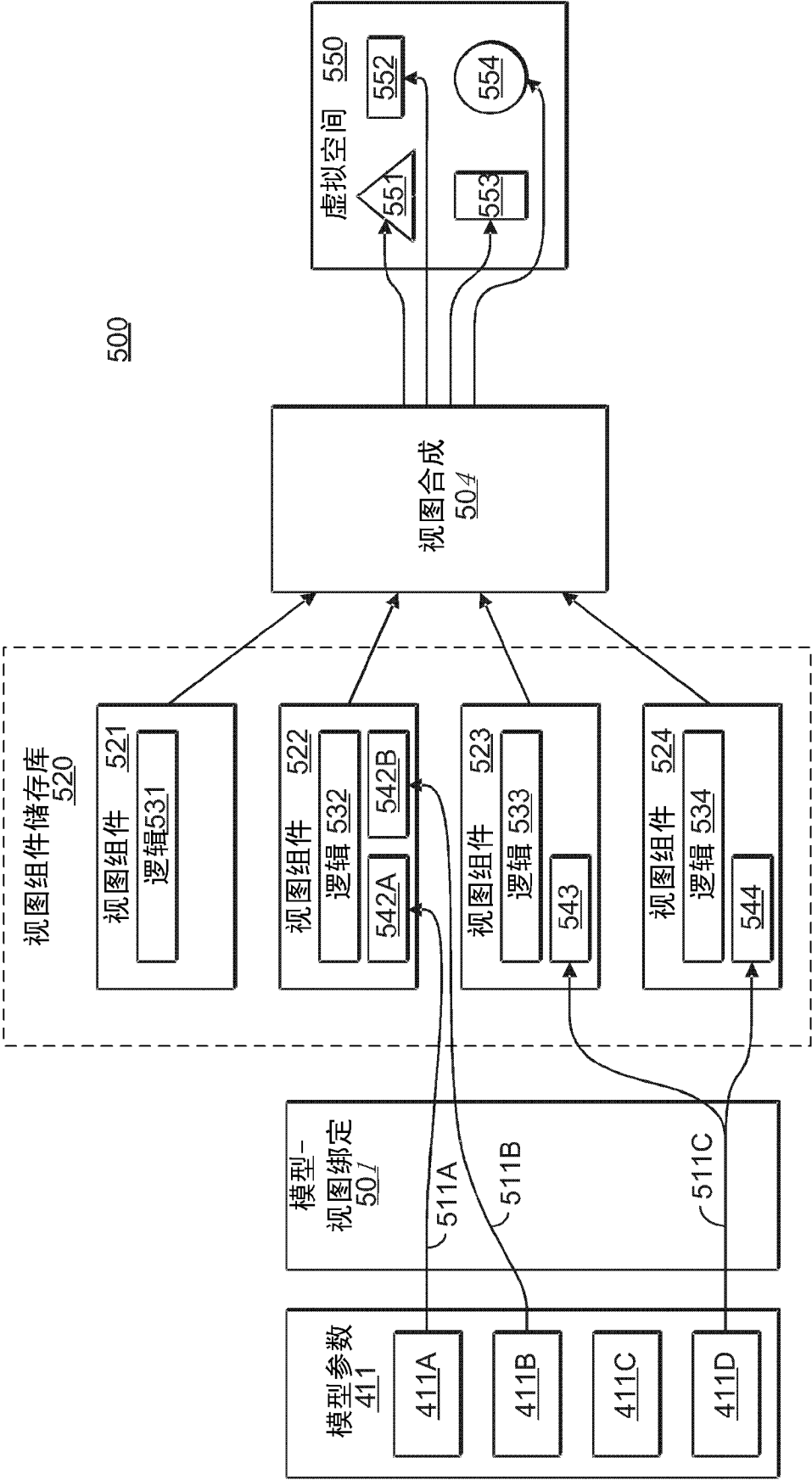


图 5

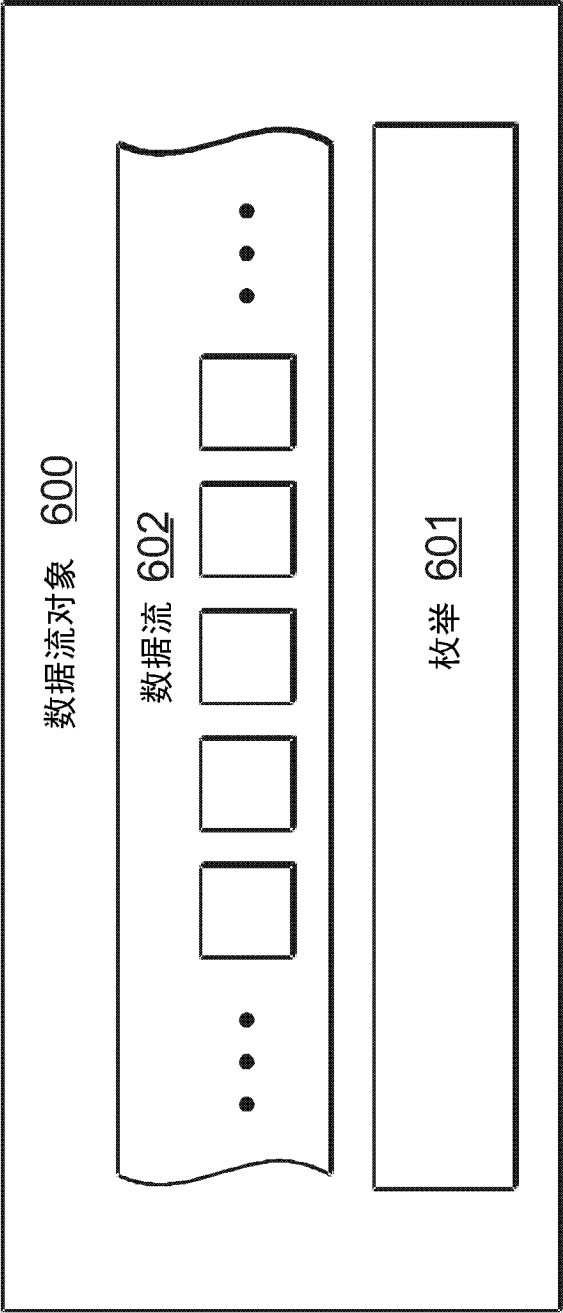


图 6

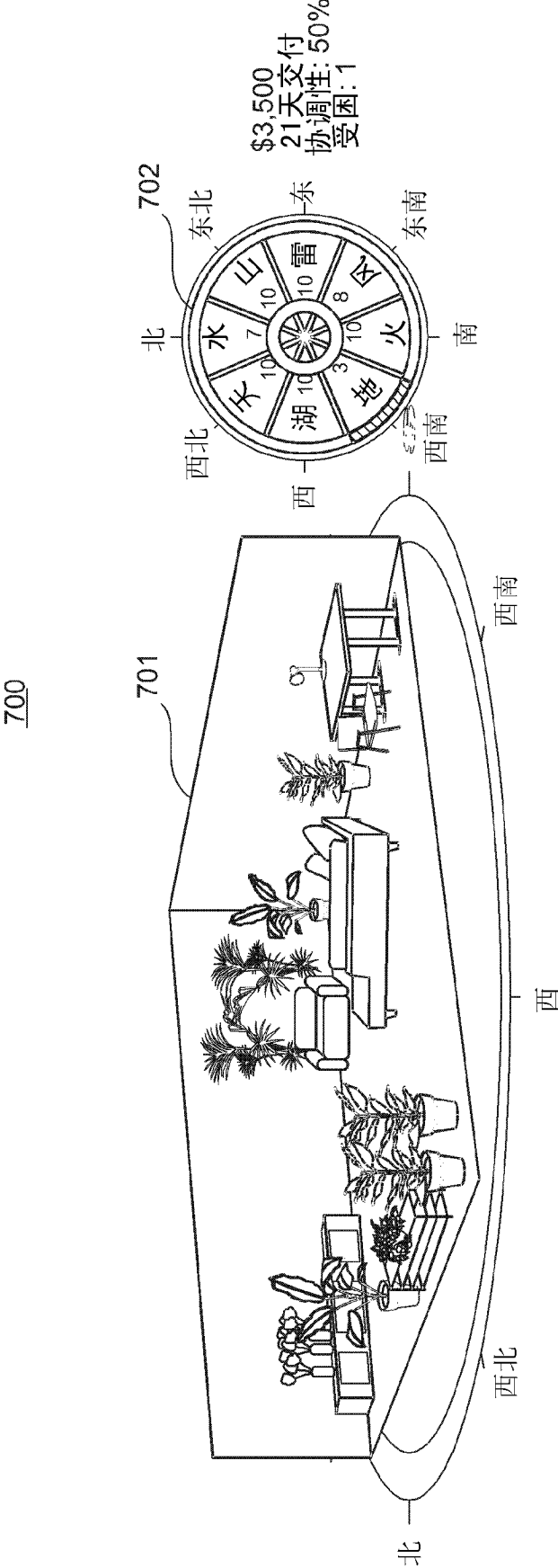


图 7

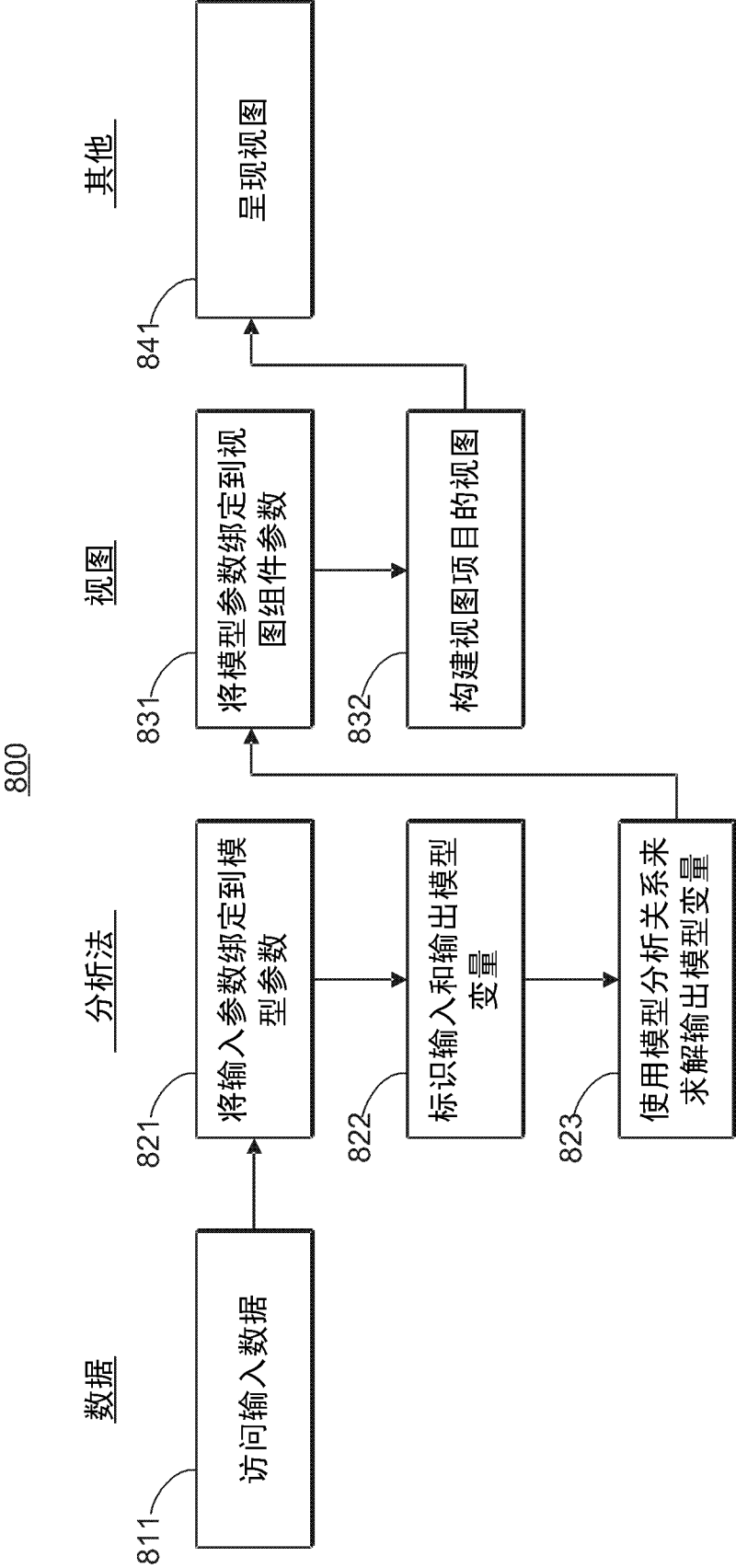


图 8

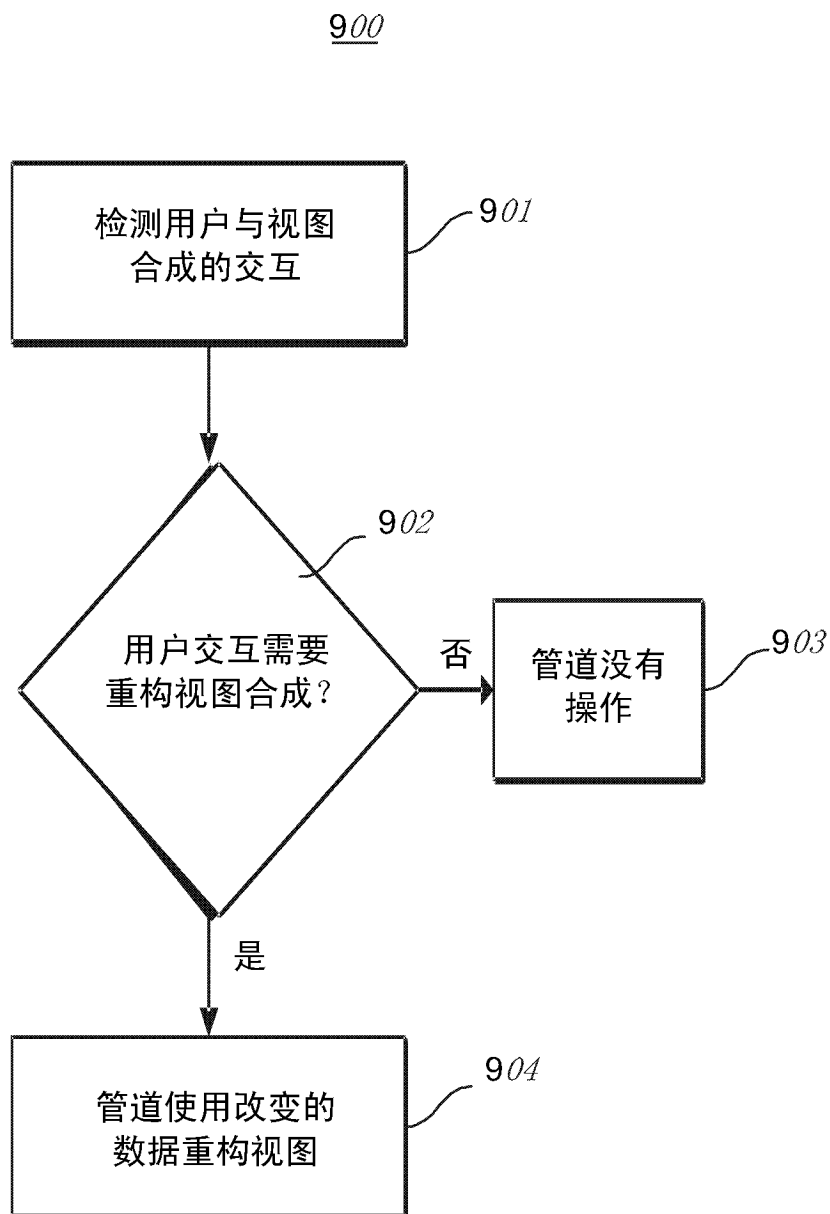


图 9

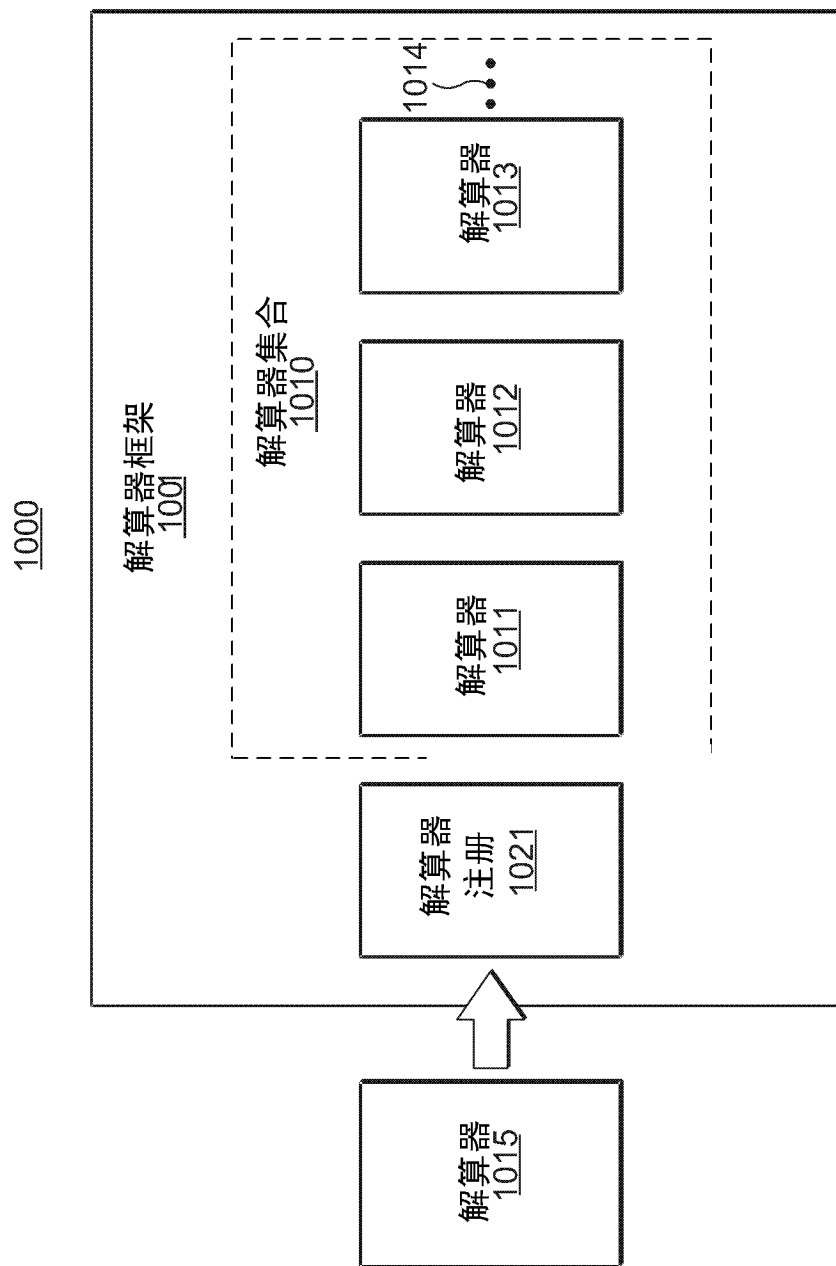


图 10

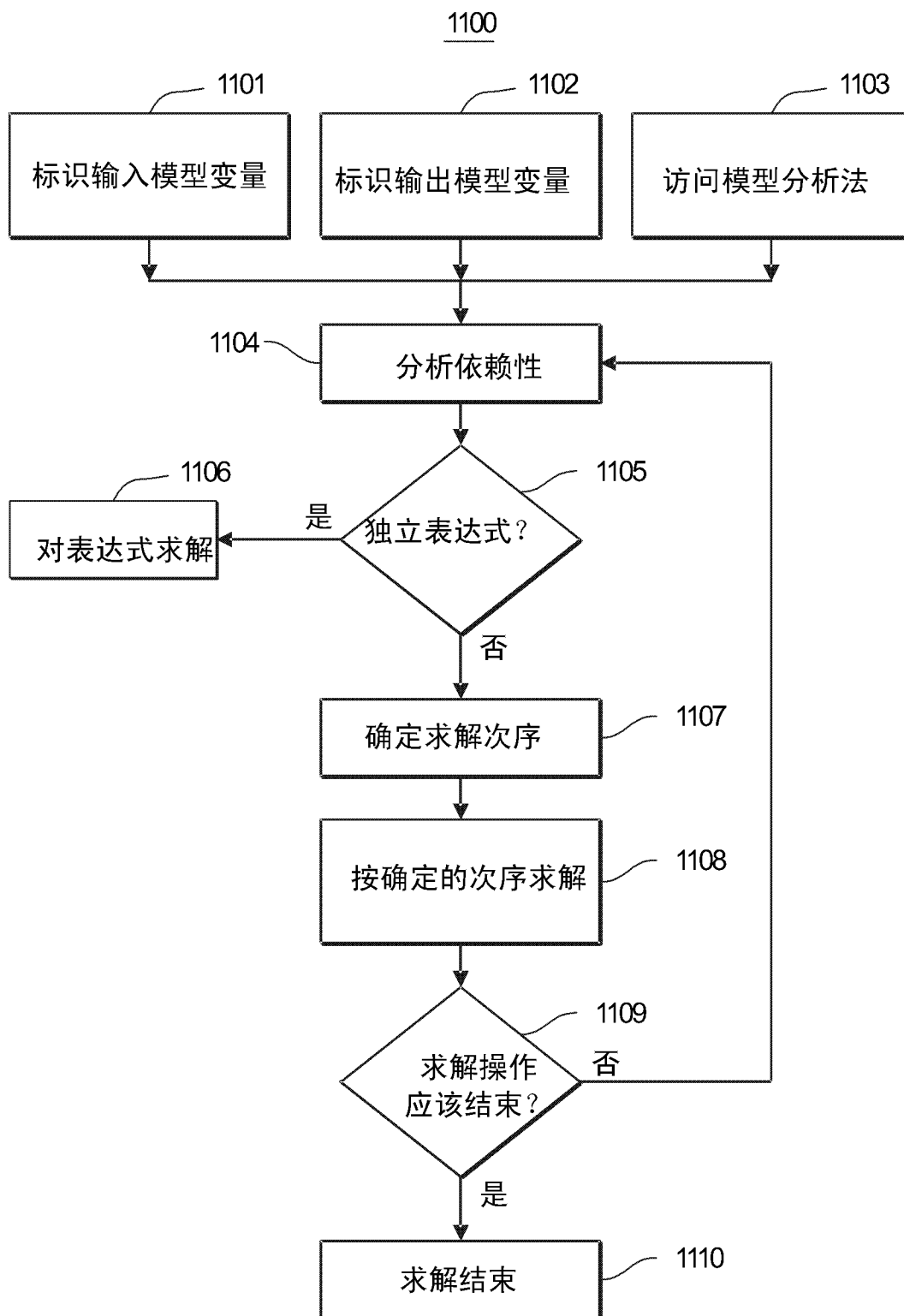


图 11

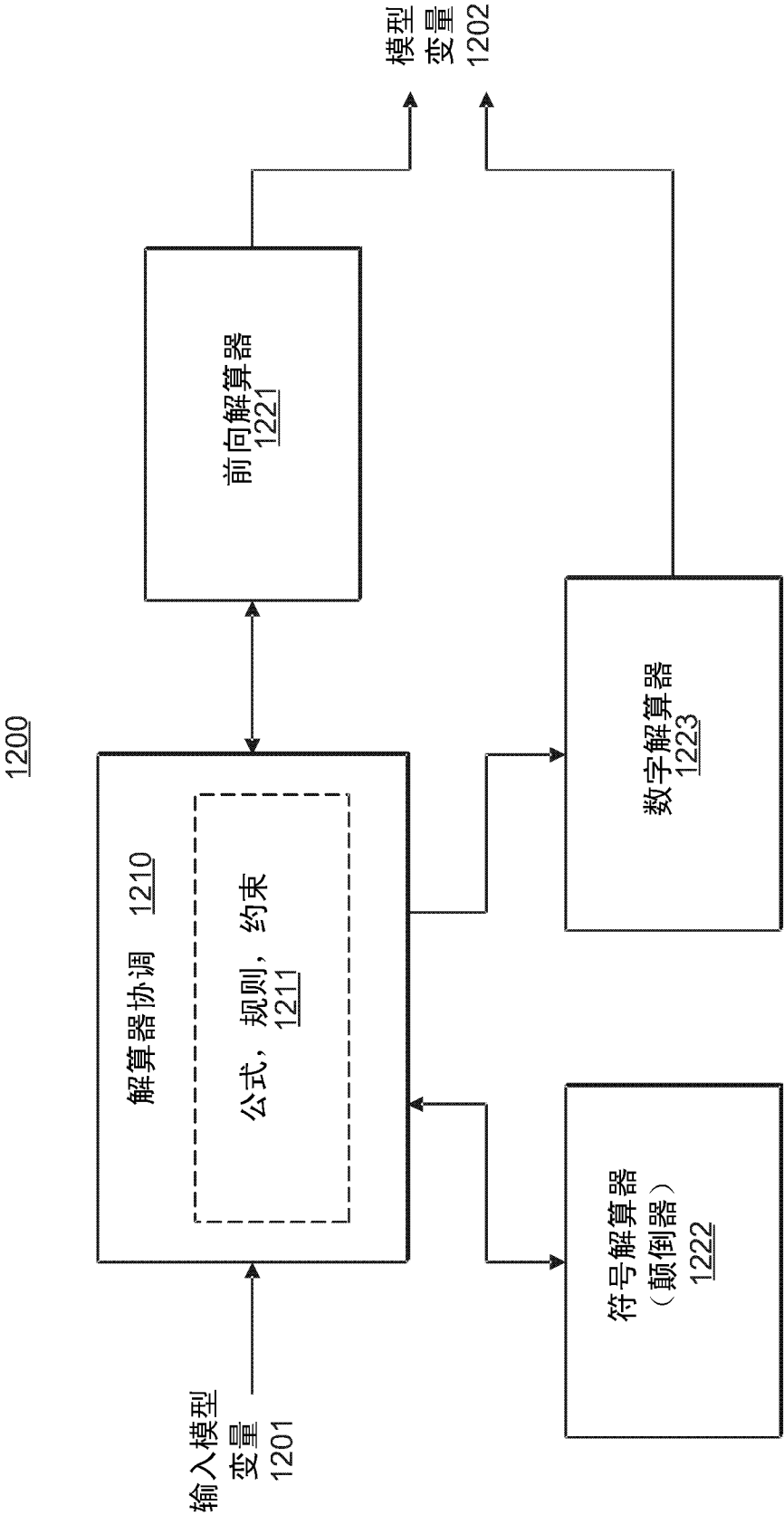


图 12

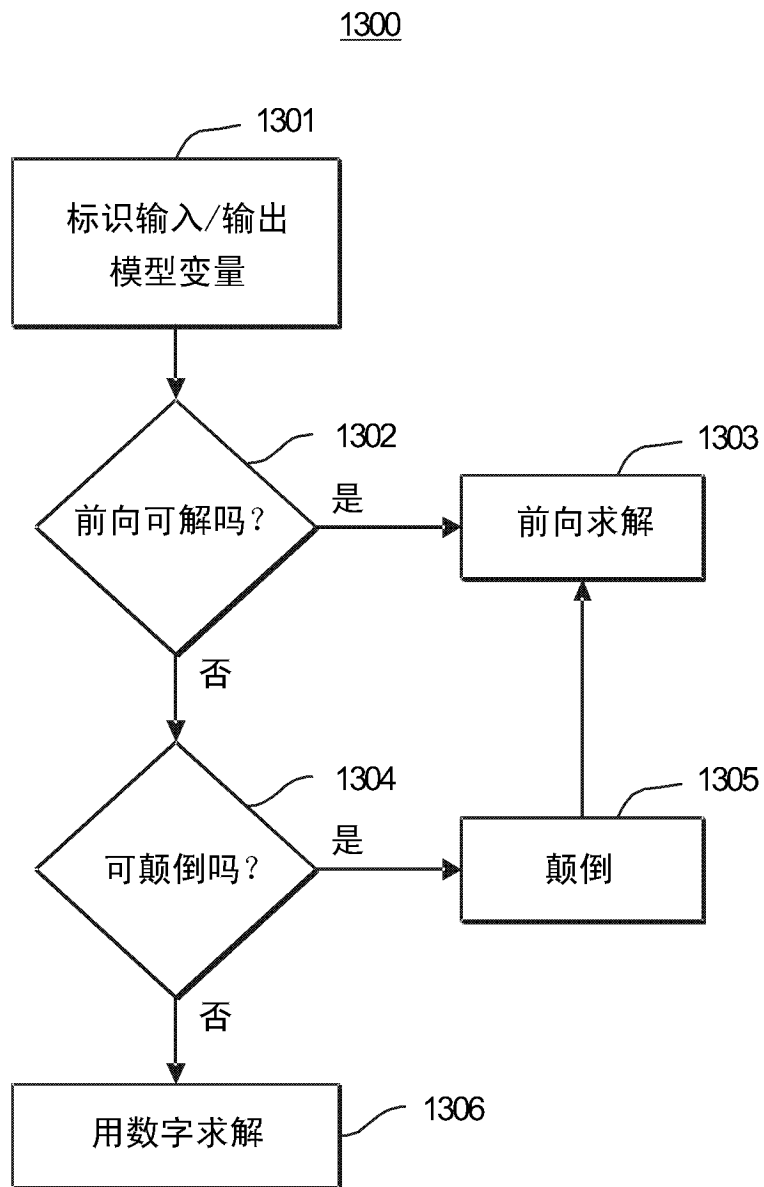


图 13

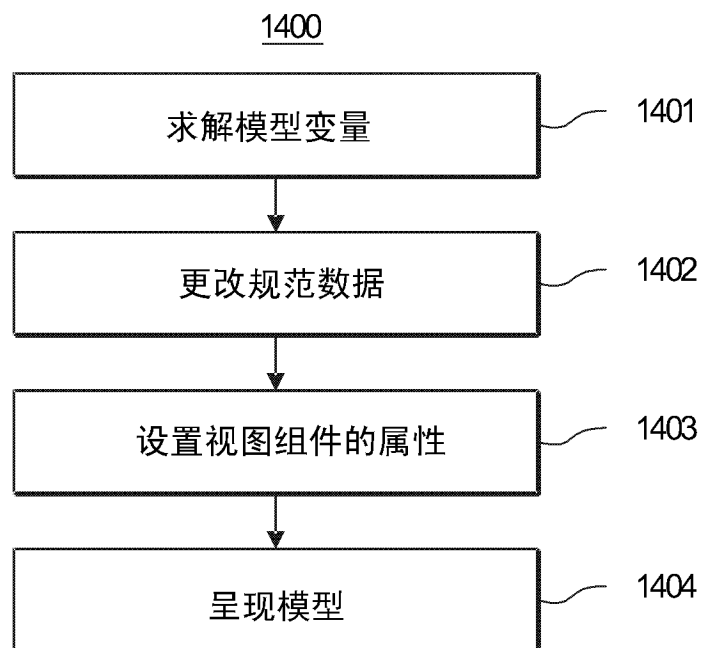


图 14

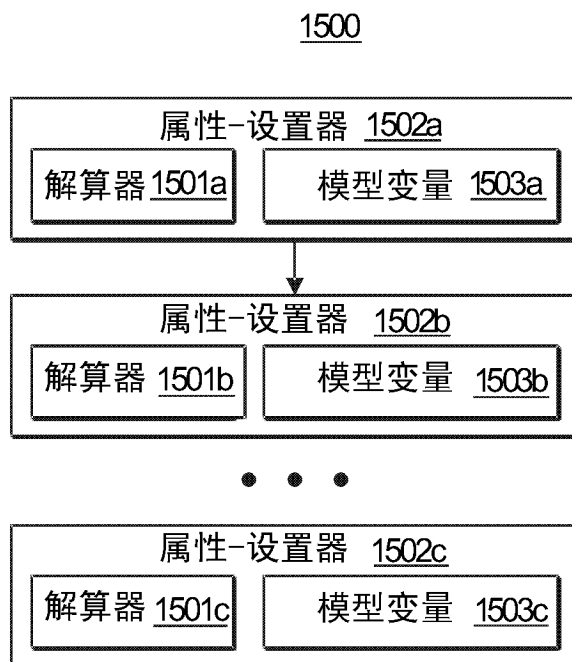


图 15

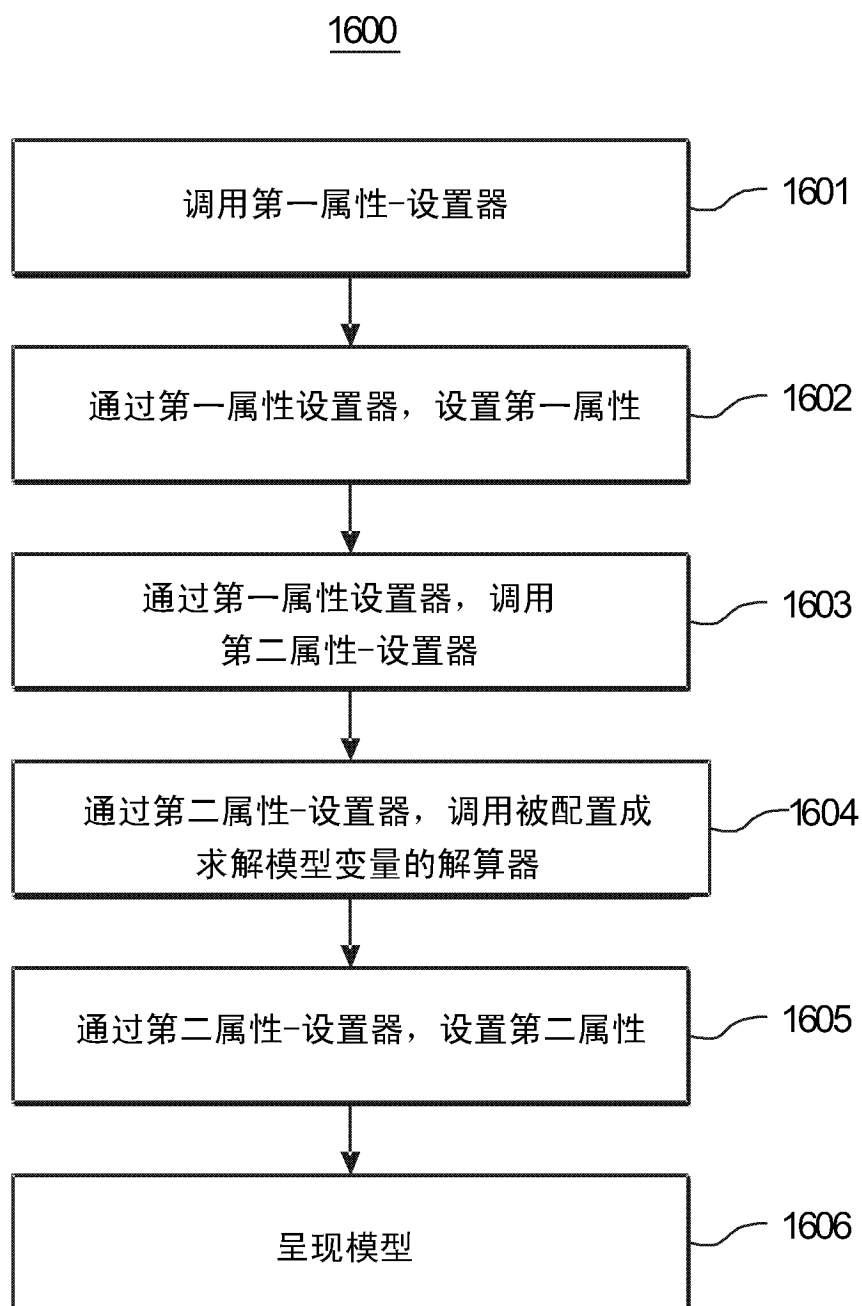


图 16

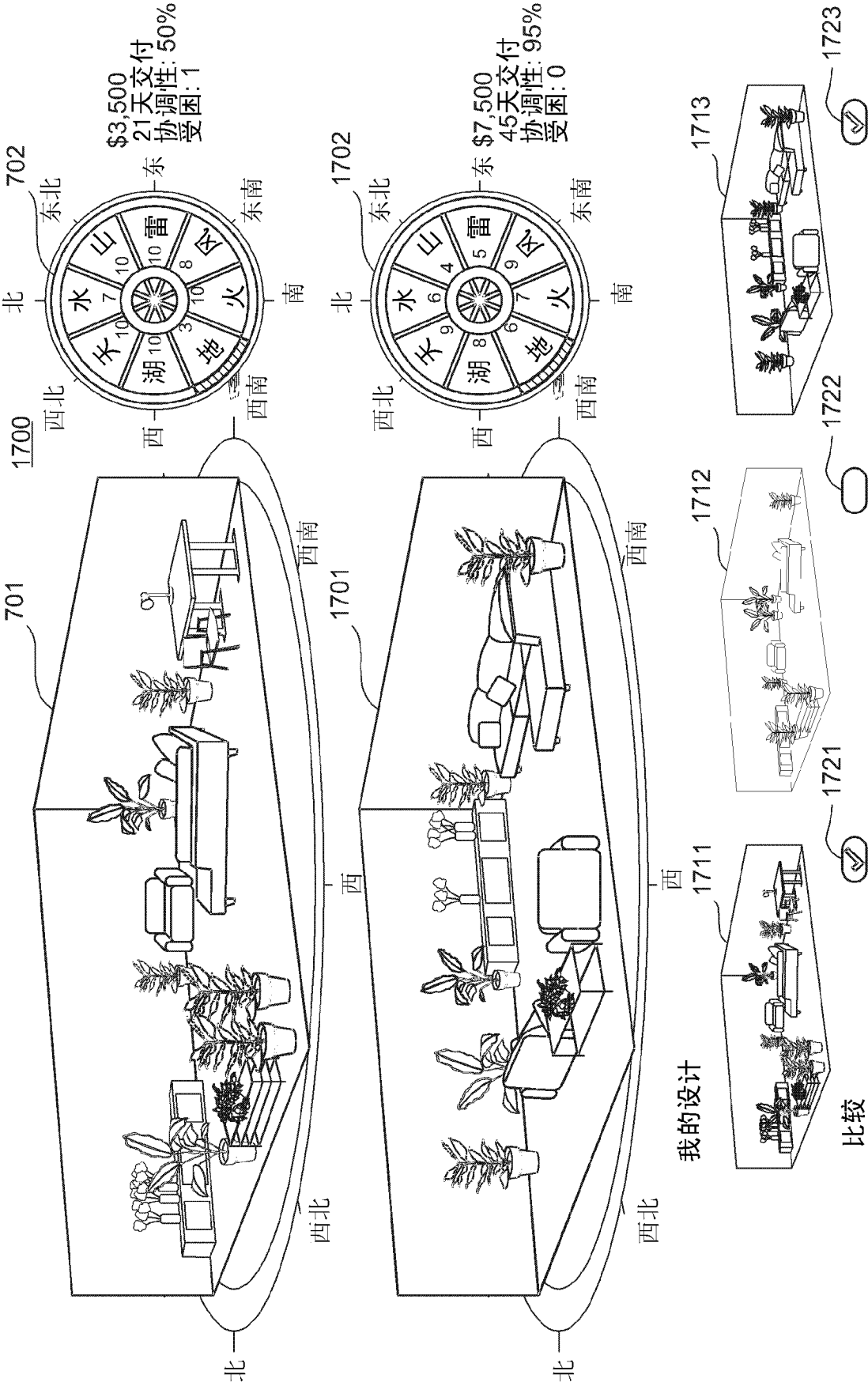


图 17

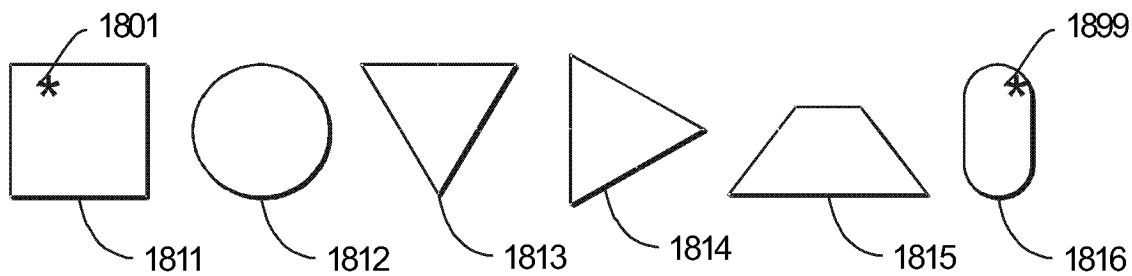
1800A

图 18A

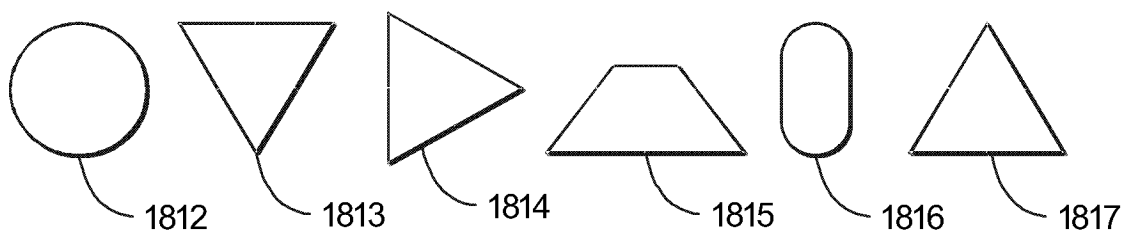
1800B

图 18B

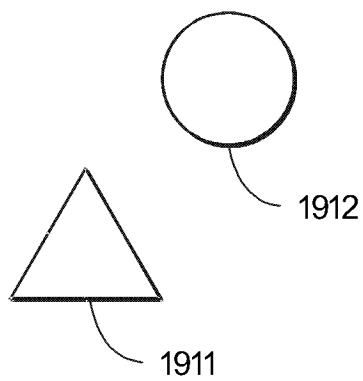
1900A

图 19A

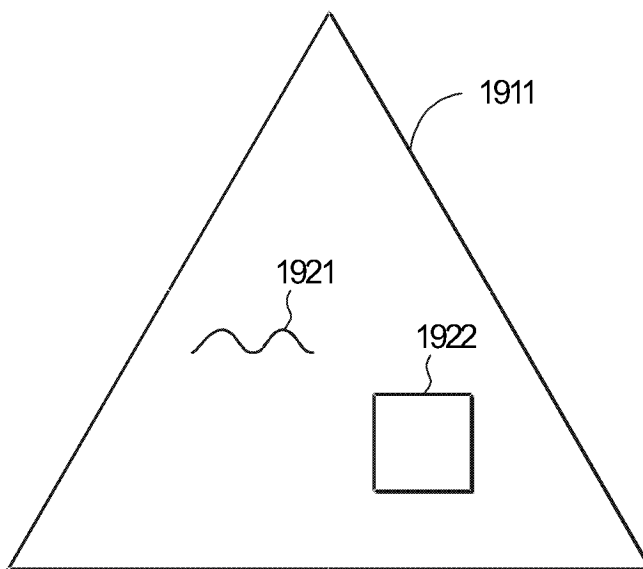
1900B

图 19B

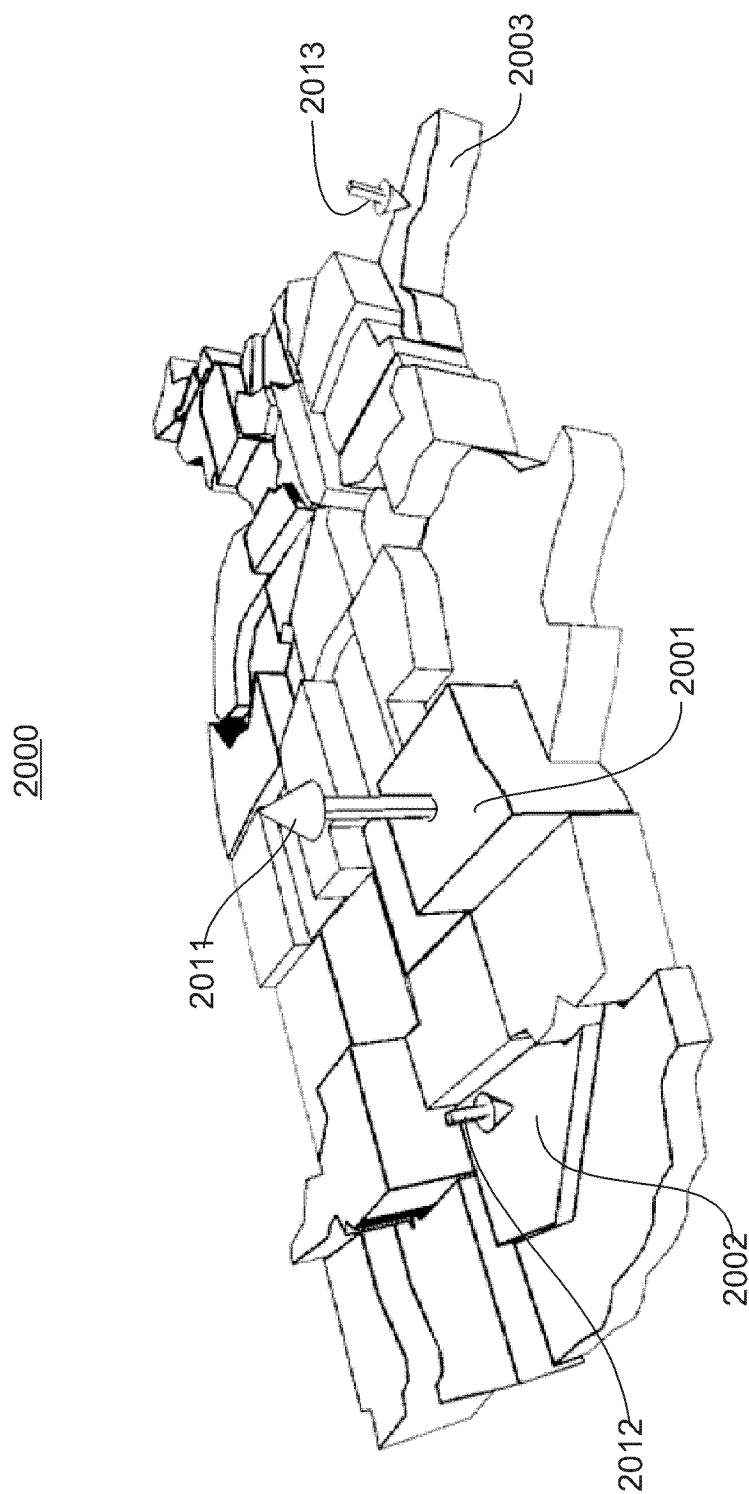


图 20

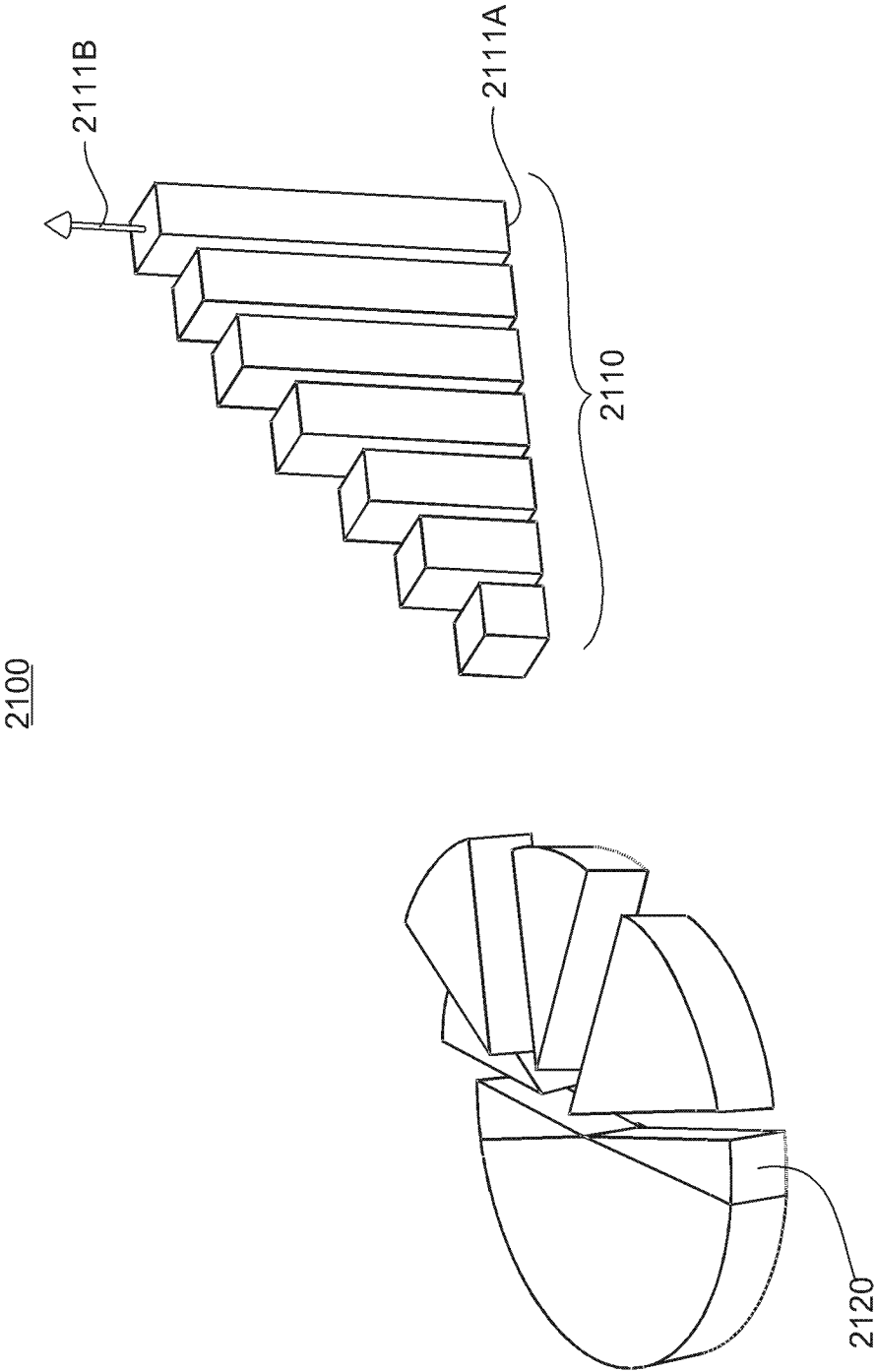


图 21

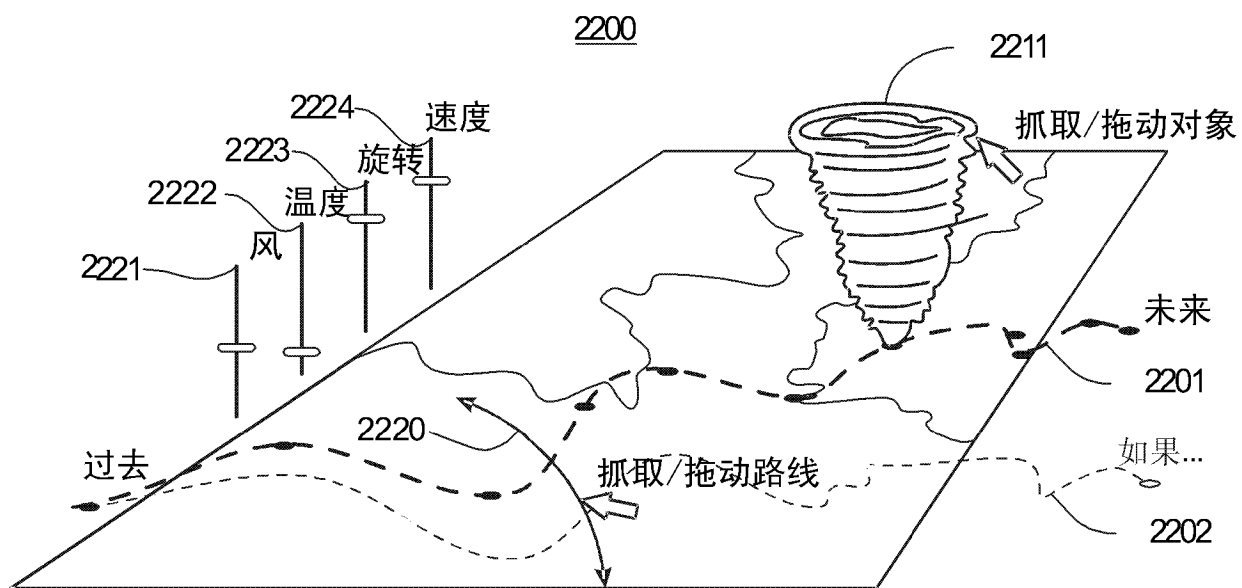


图 22

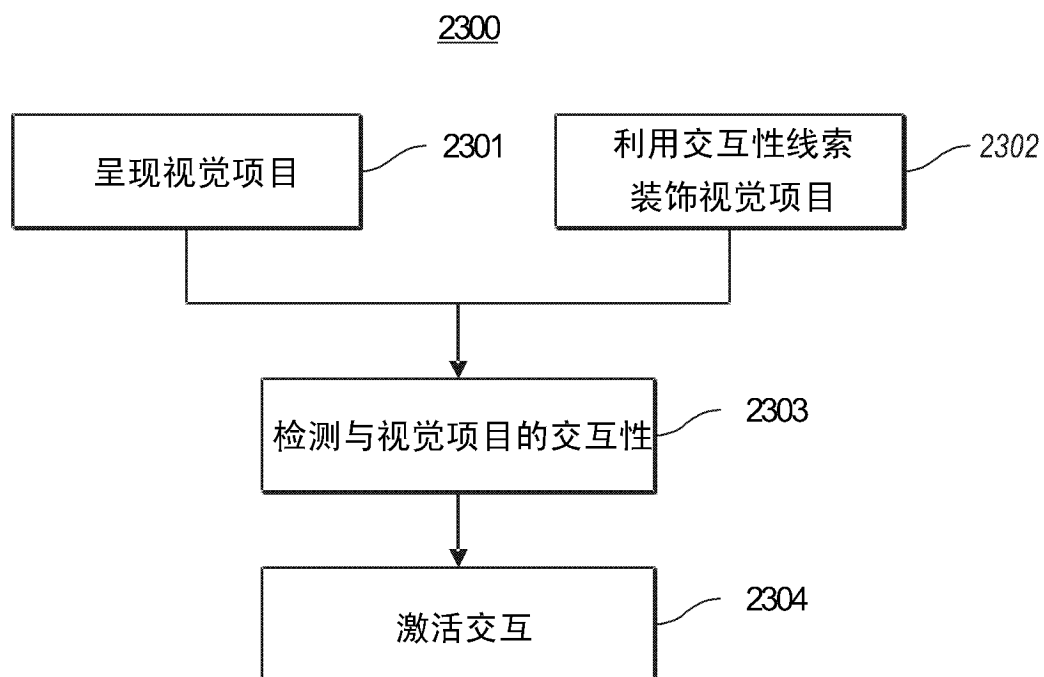


图 23

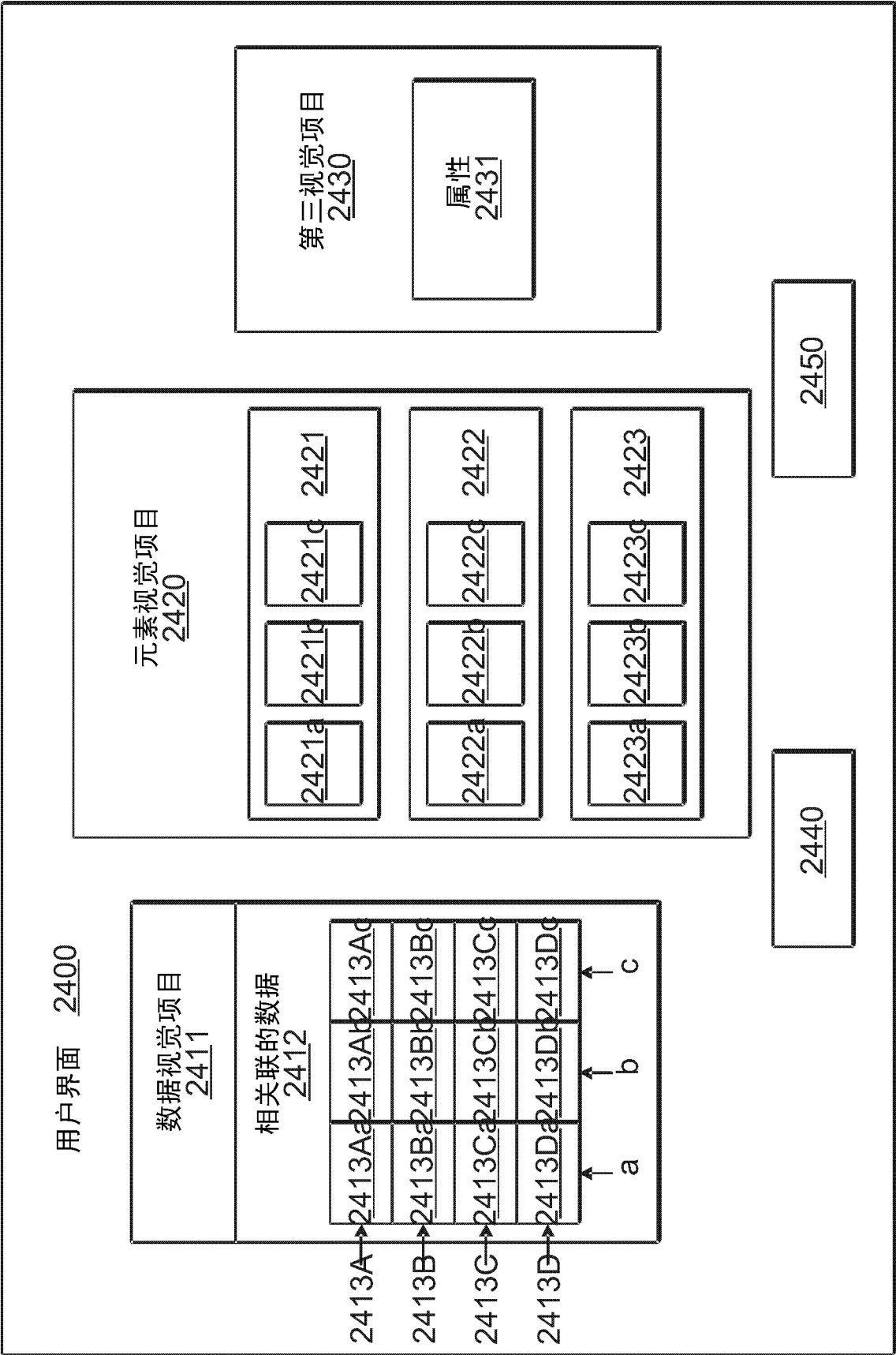


图 24

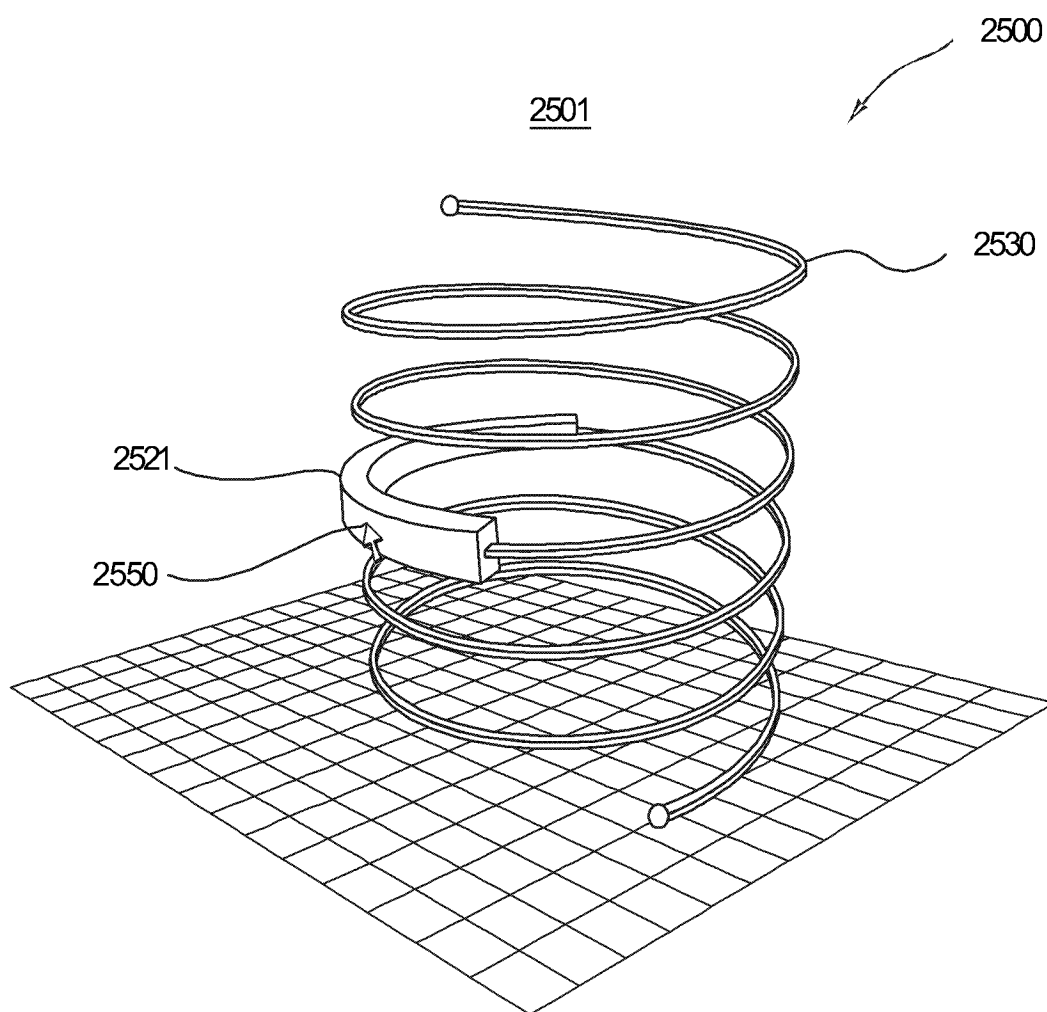


图 25

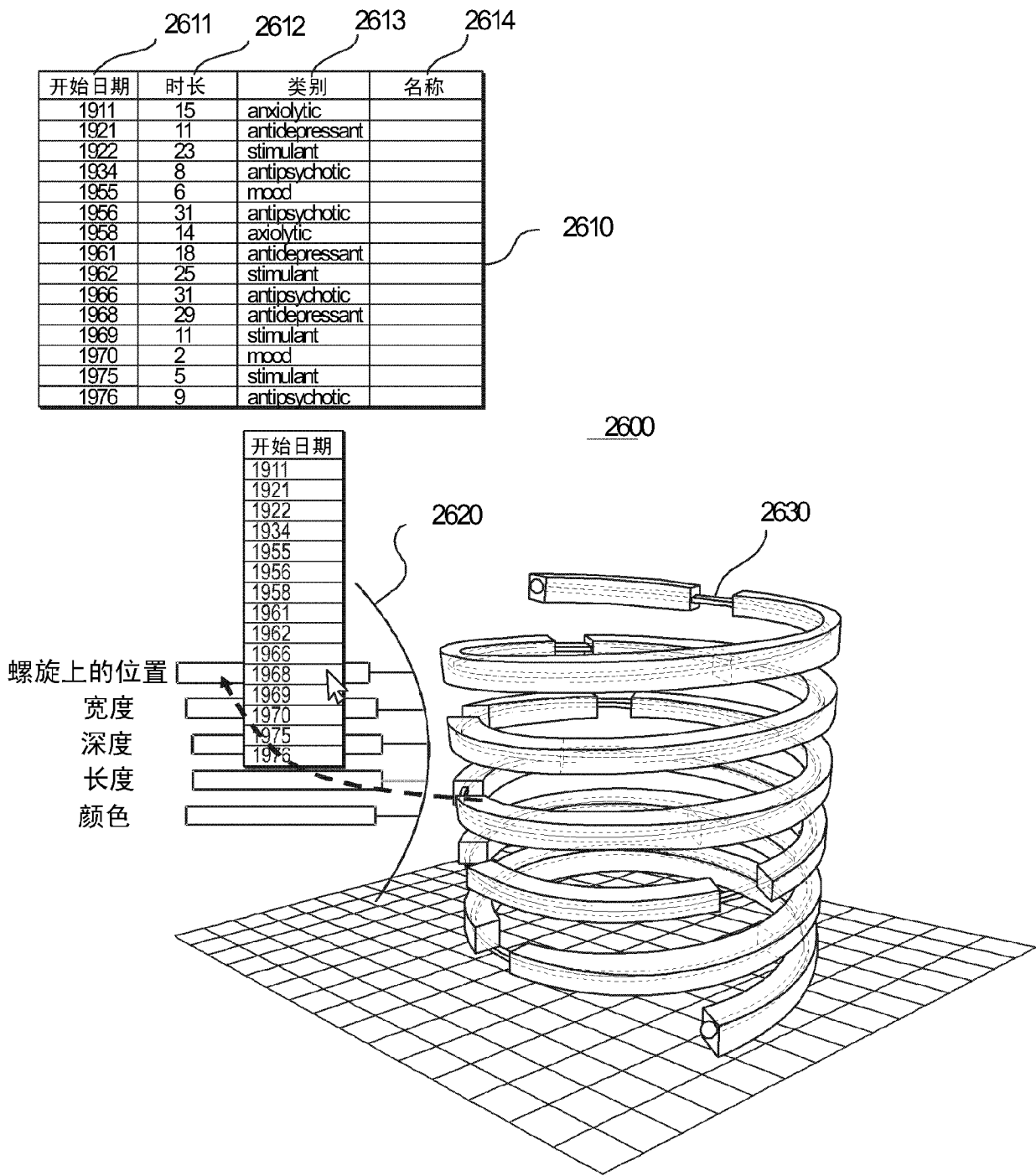


图 26

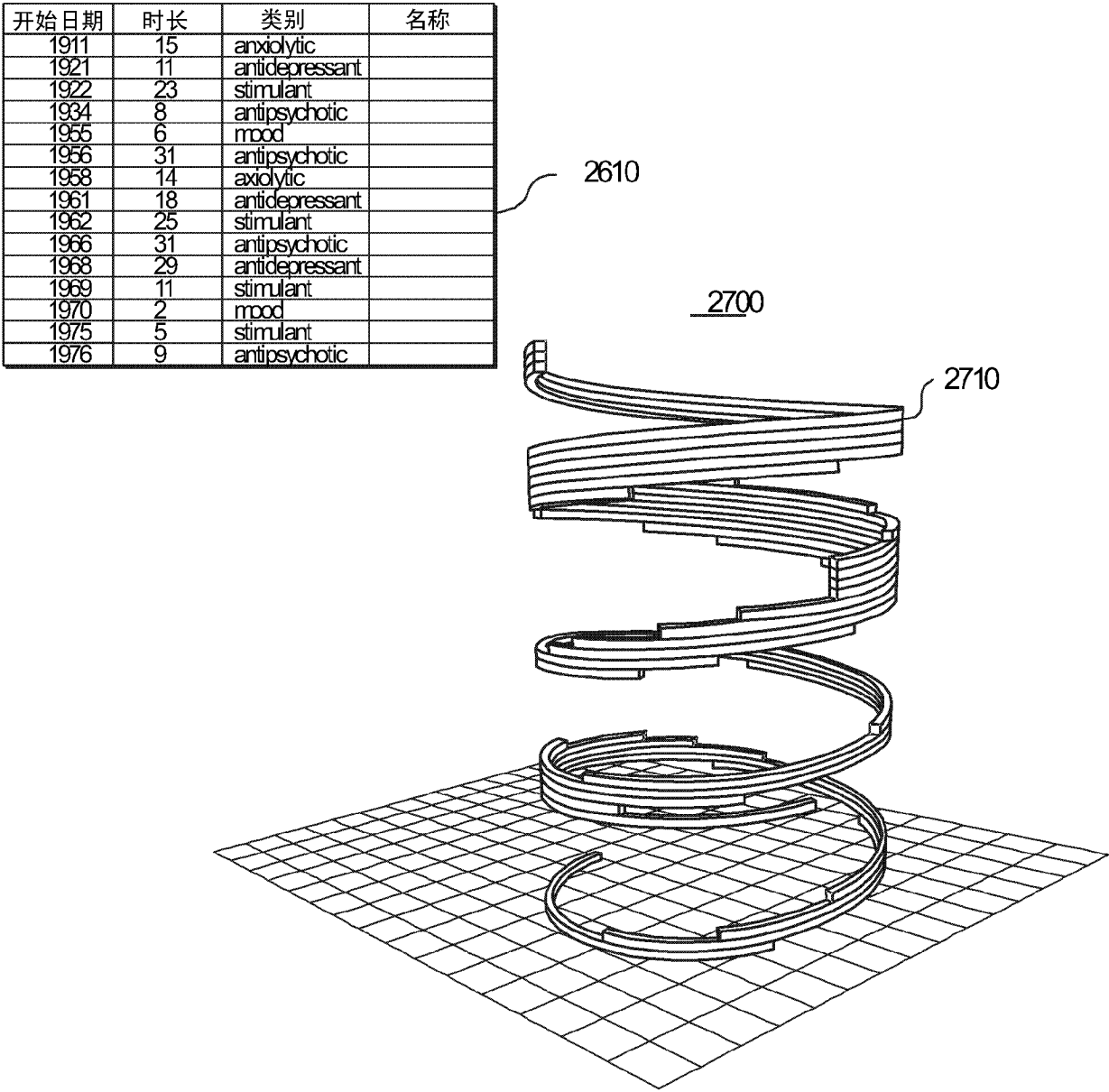


图 27

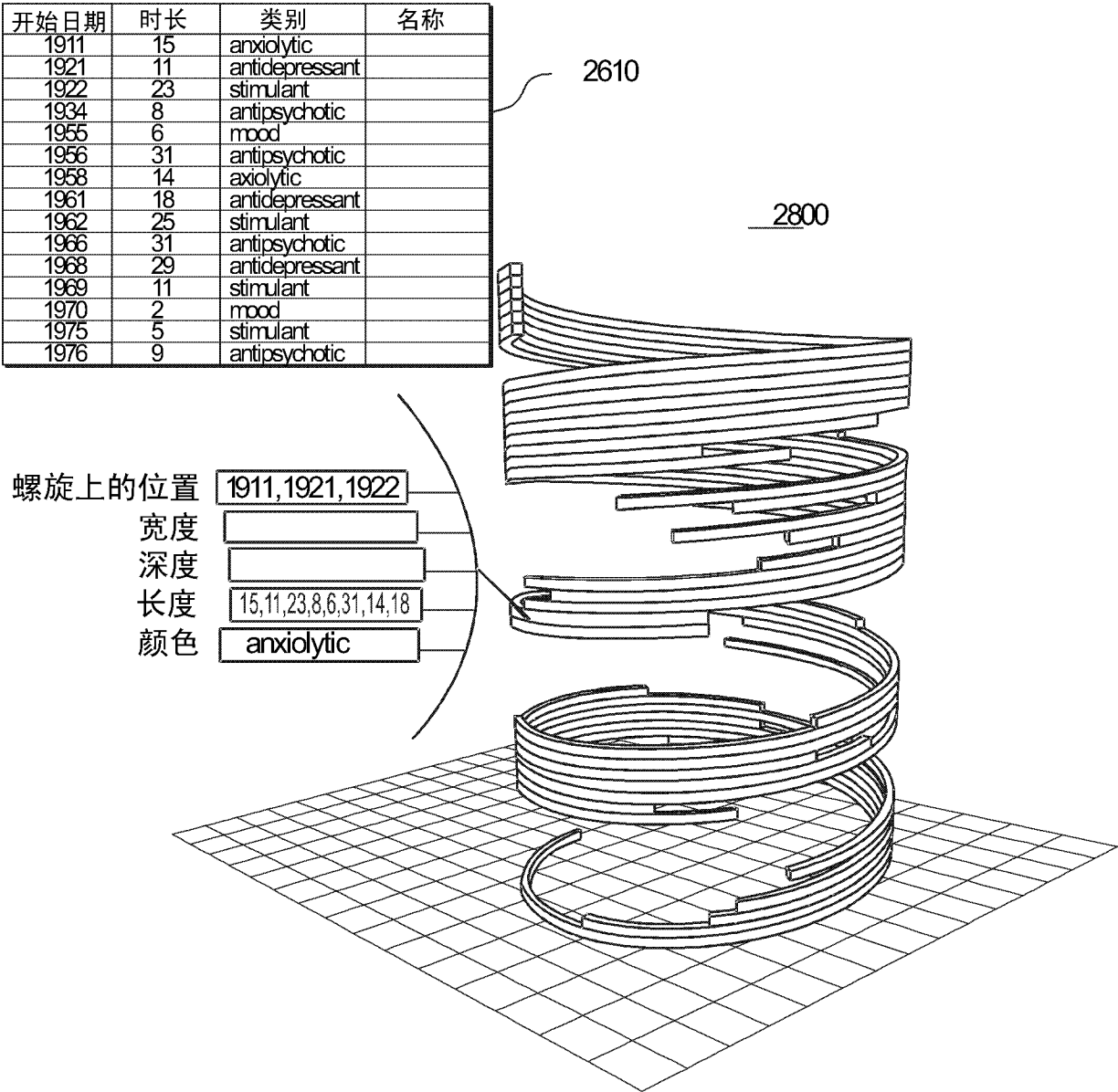


图 28

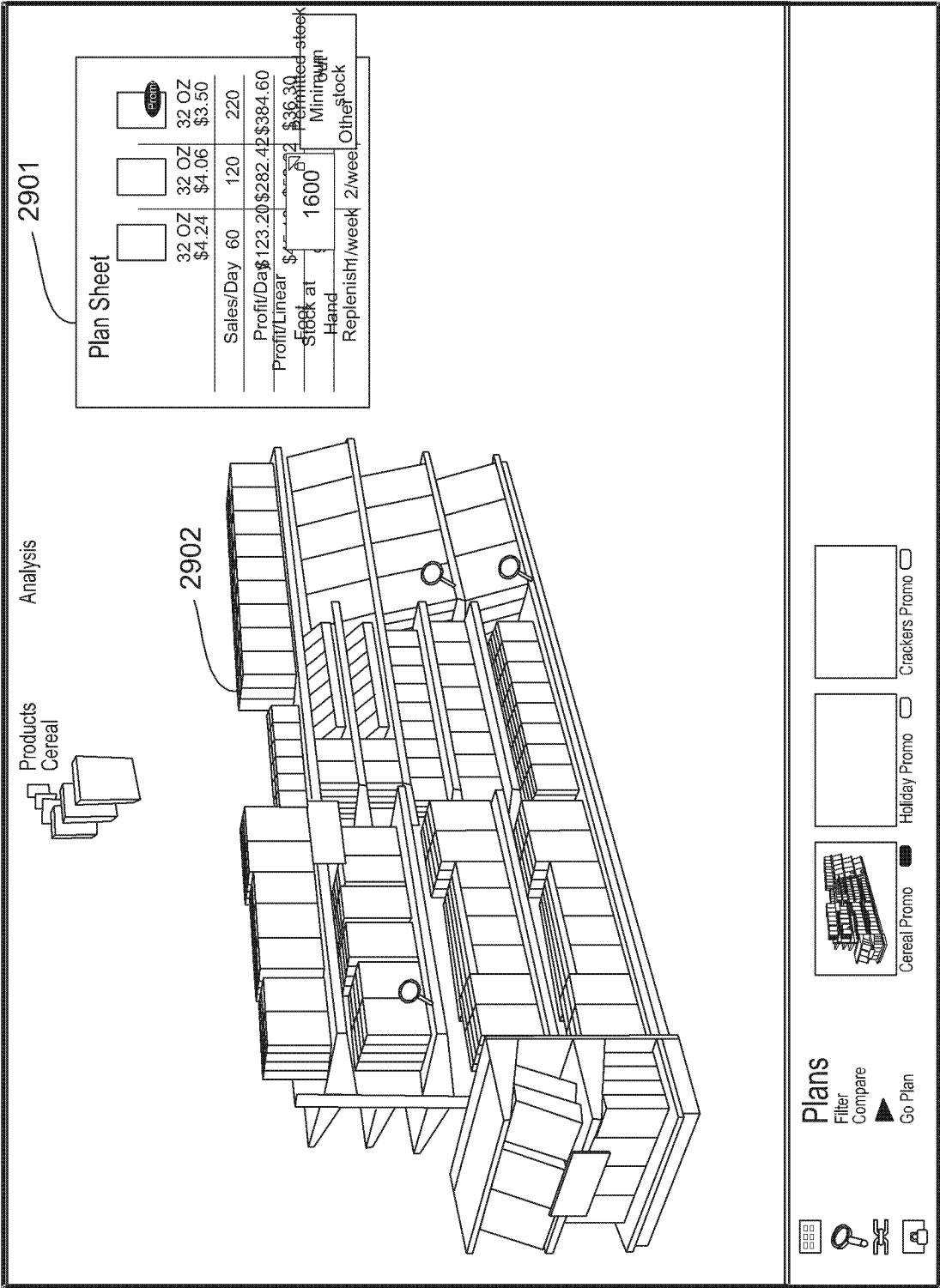


图 29

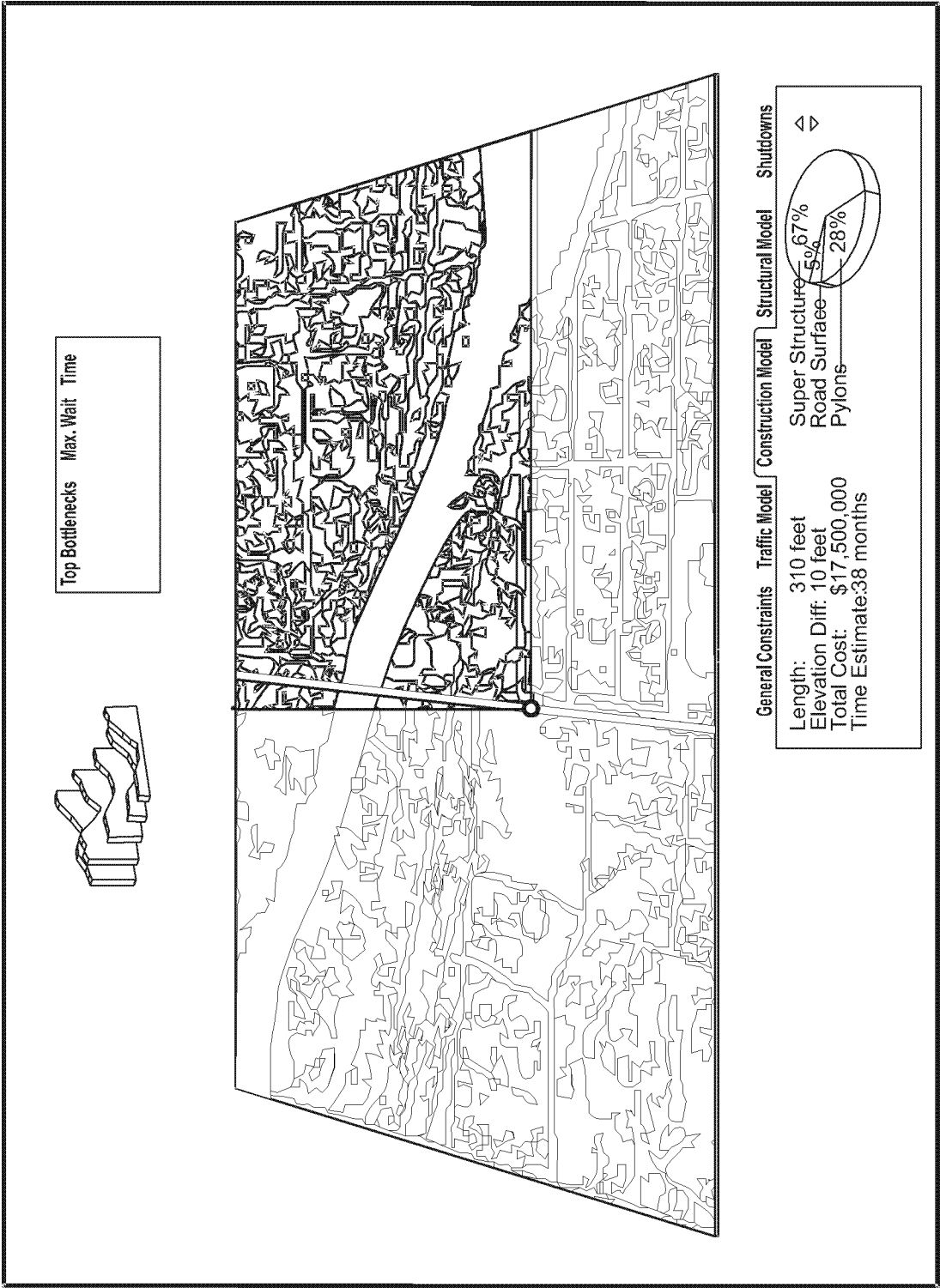


图 30

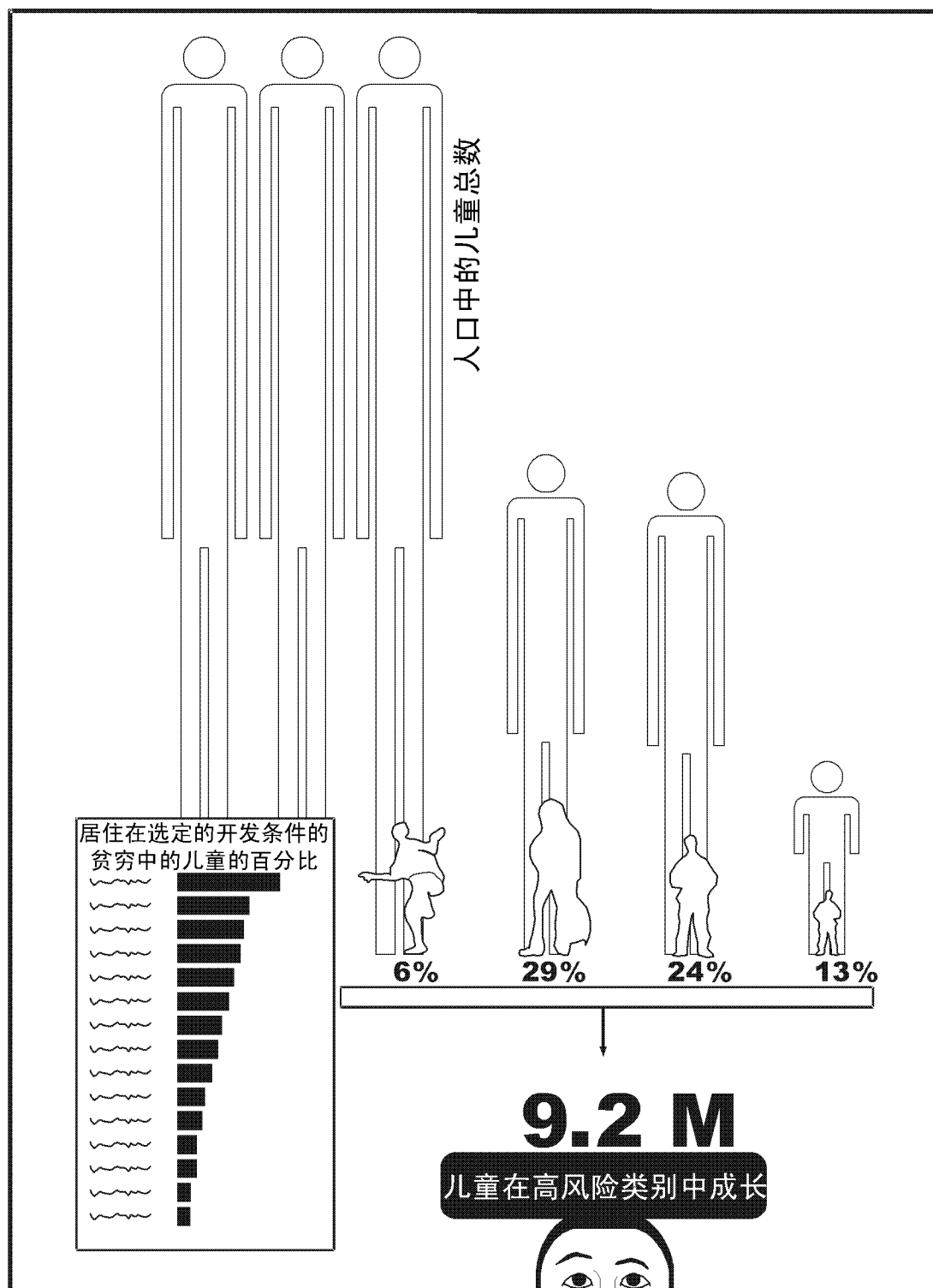


图 31

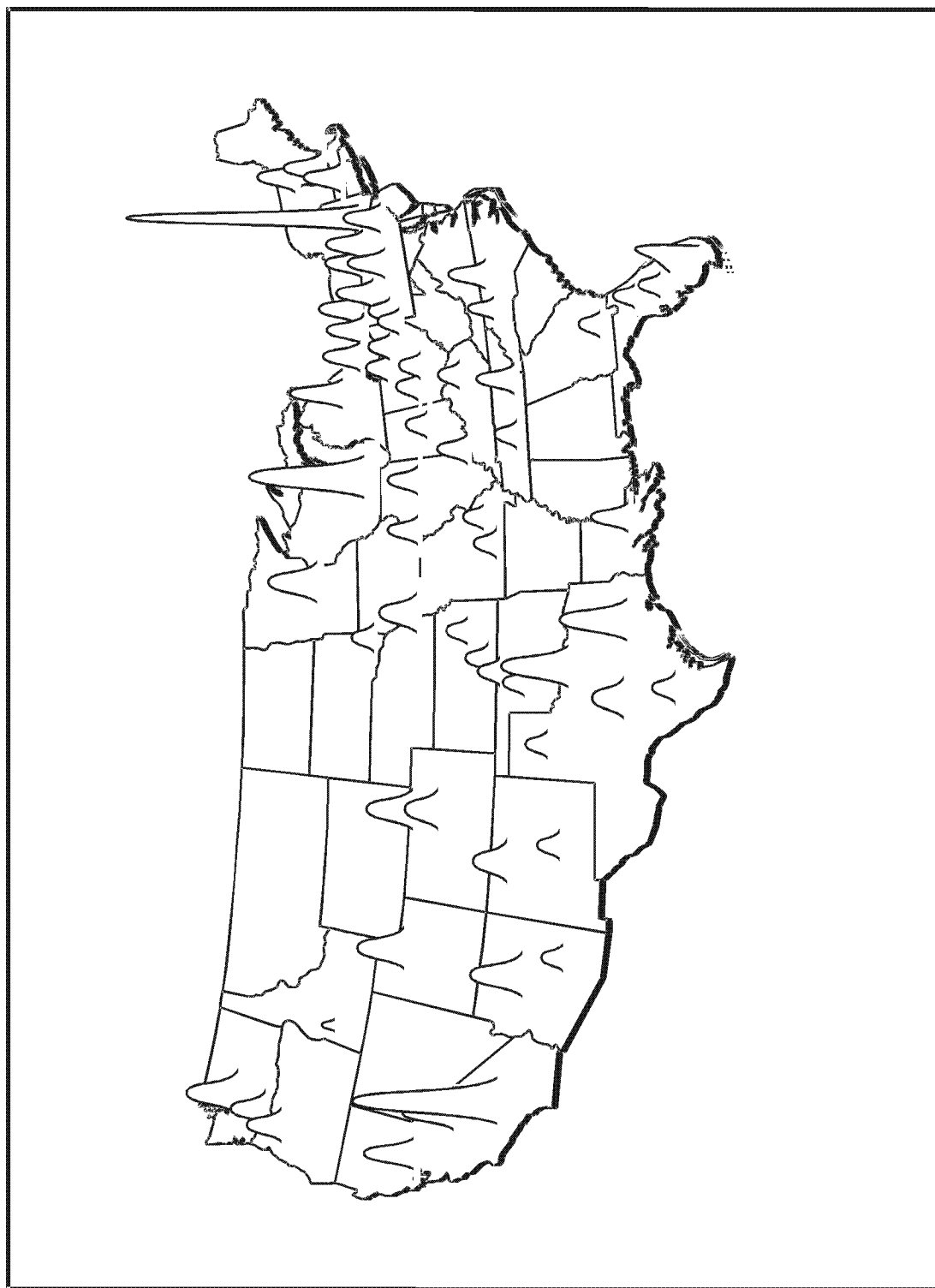


图 32

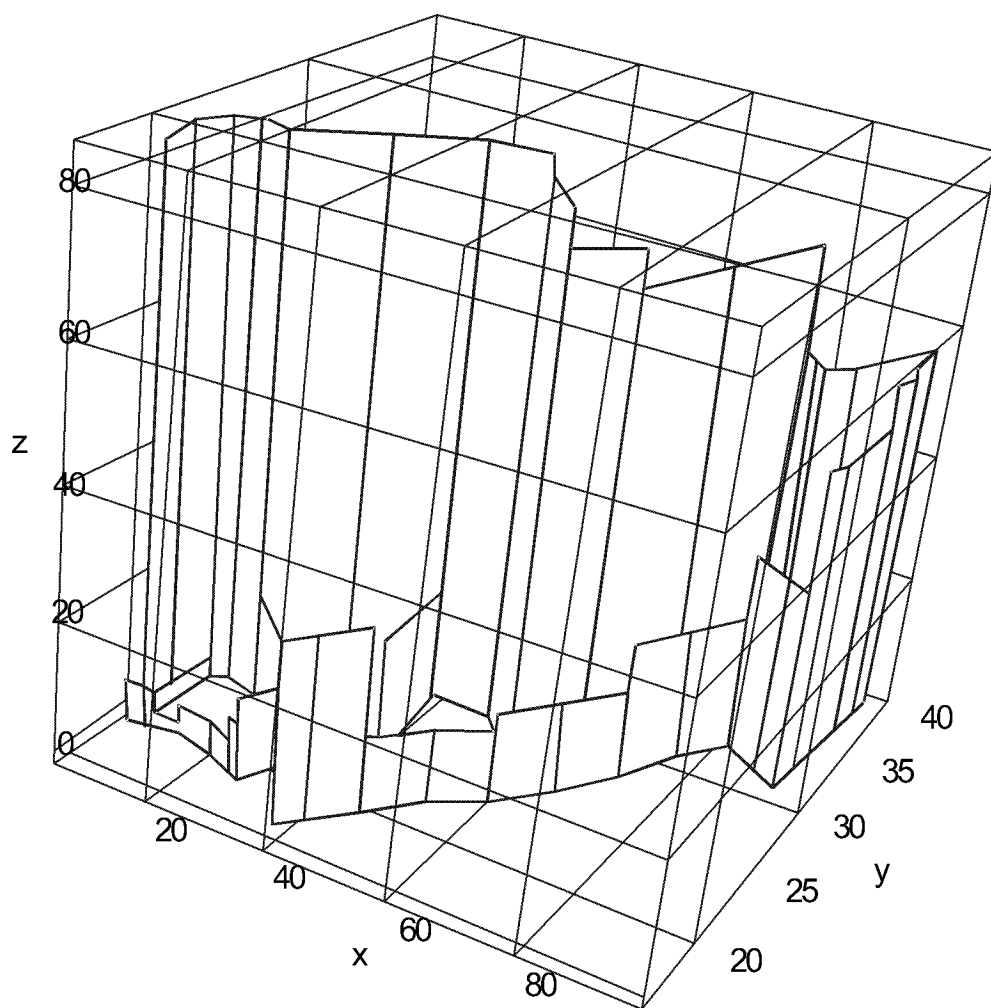


图 33

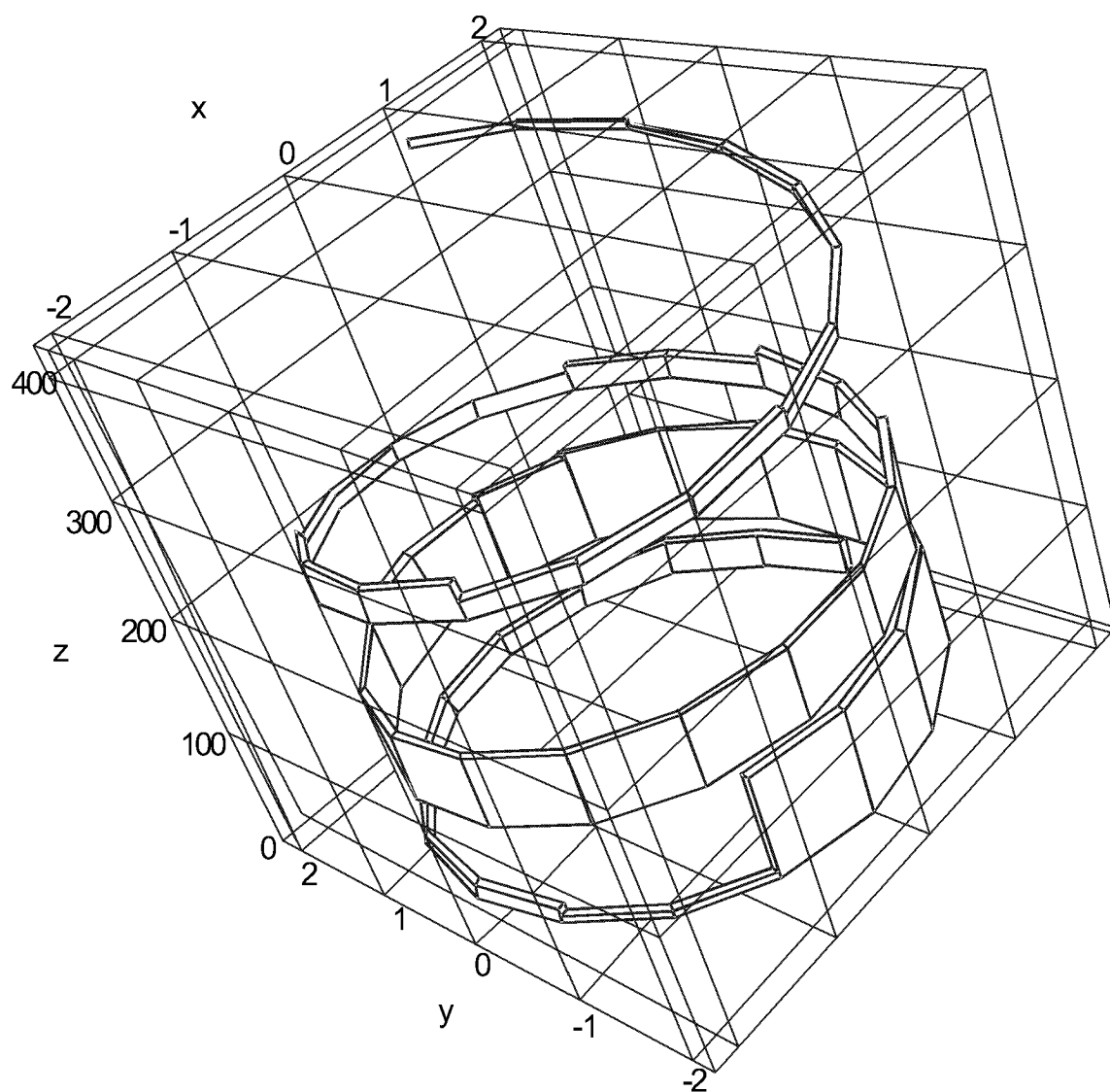


图 34

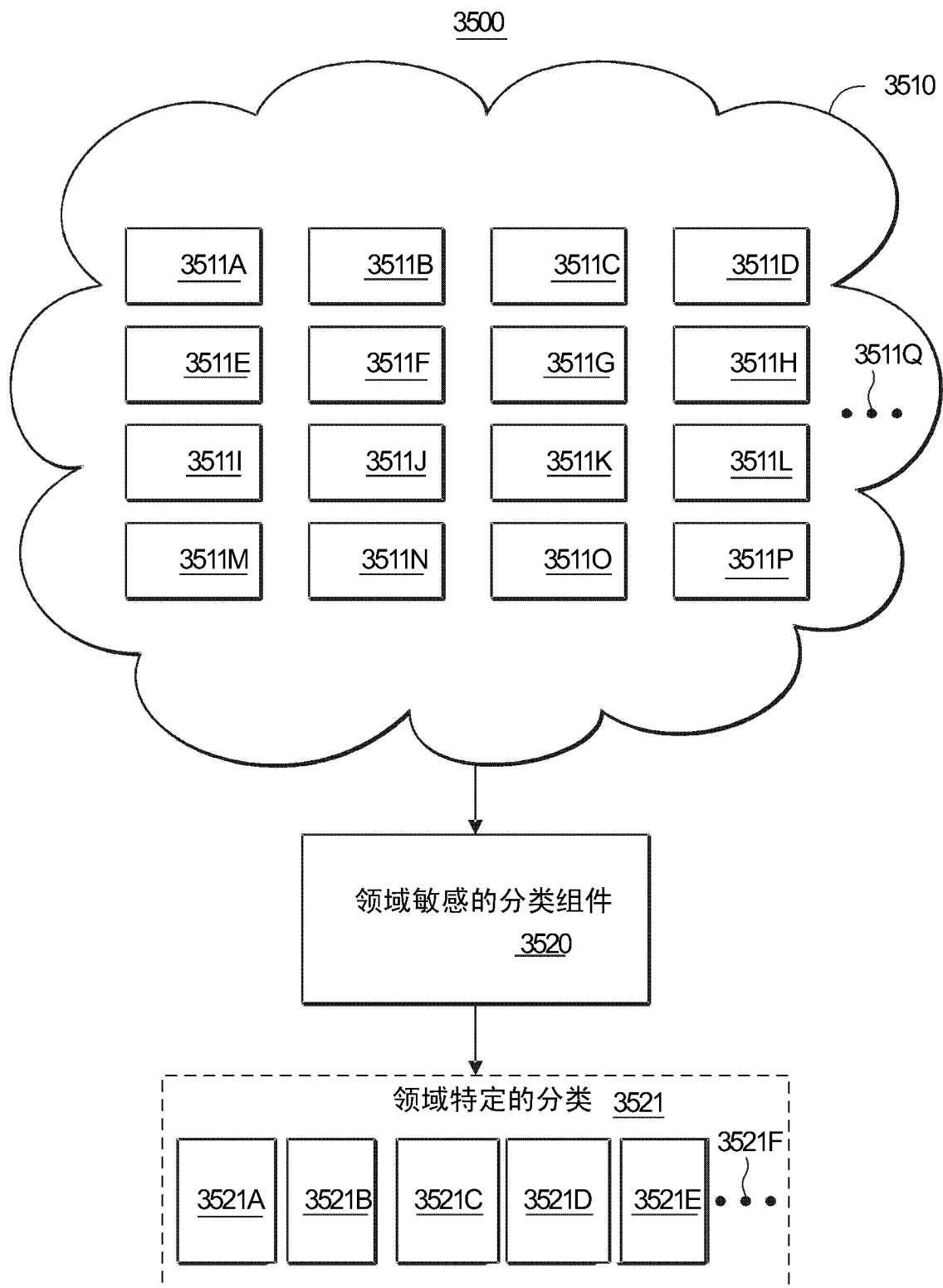


图 35

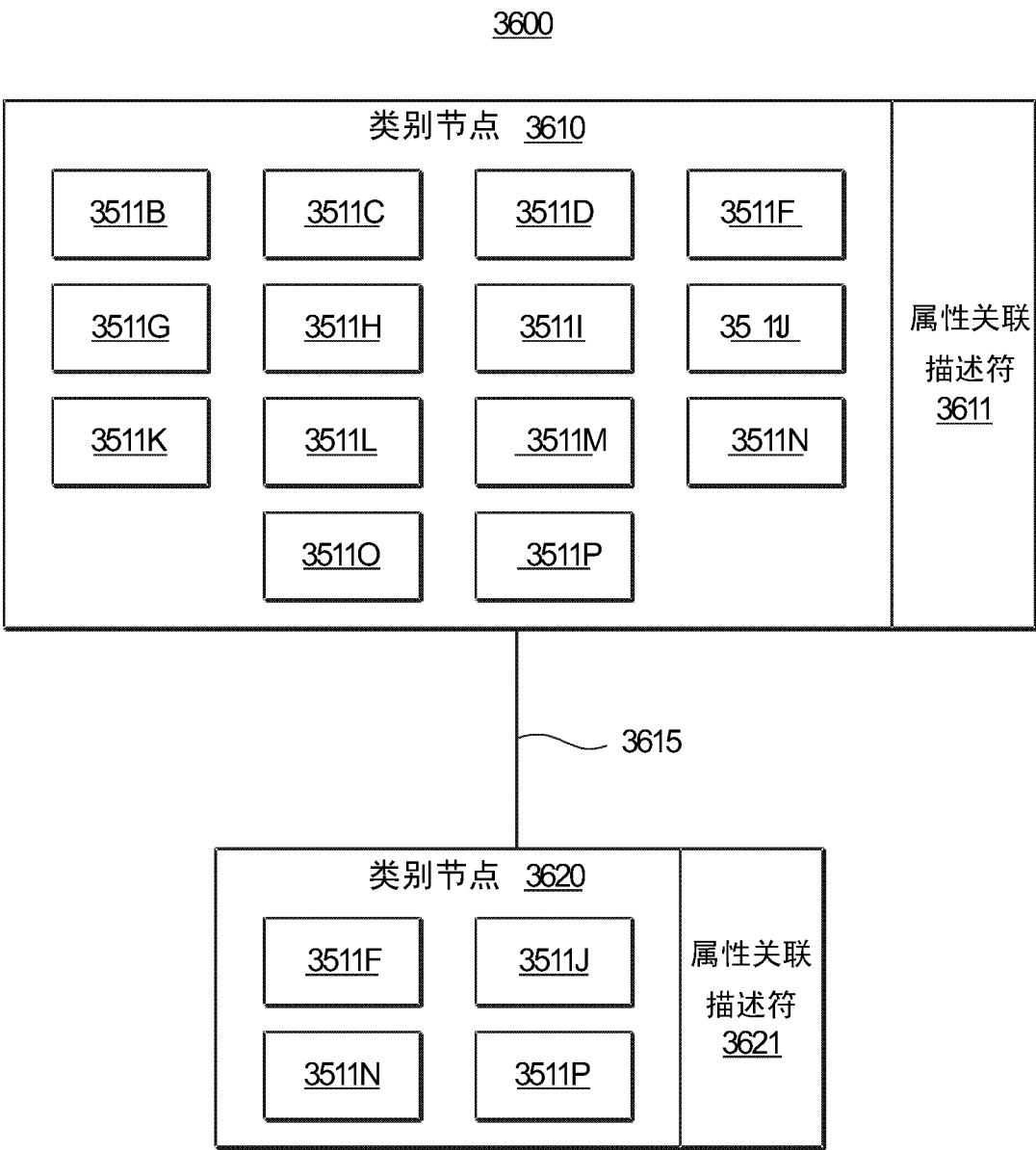


图 36

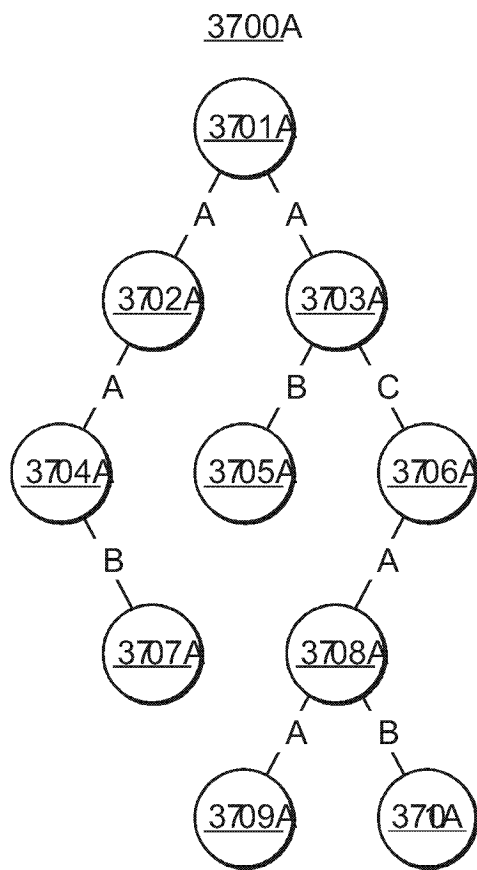


图 37A

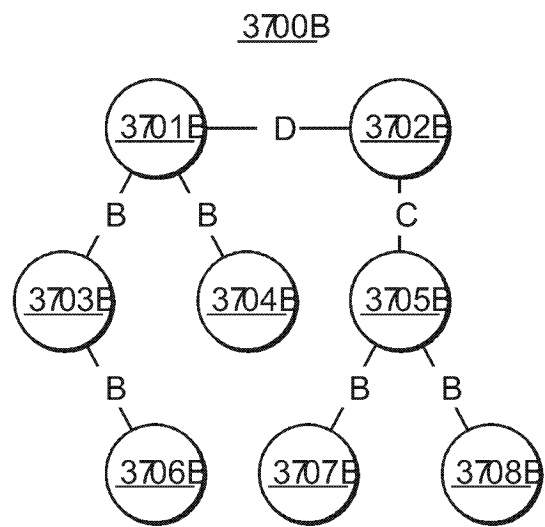


图 37B

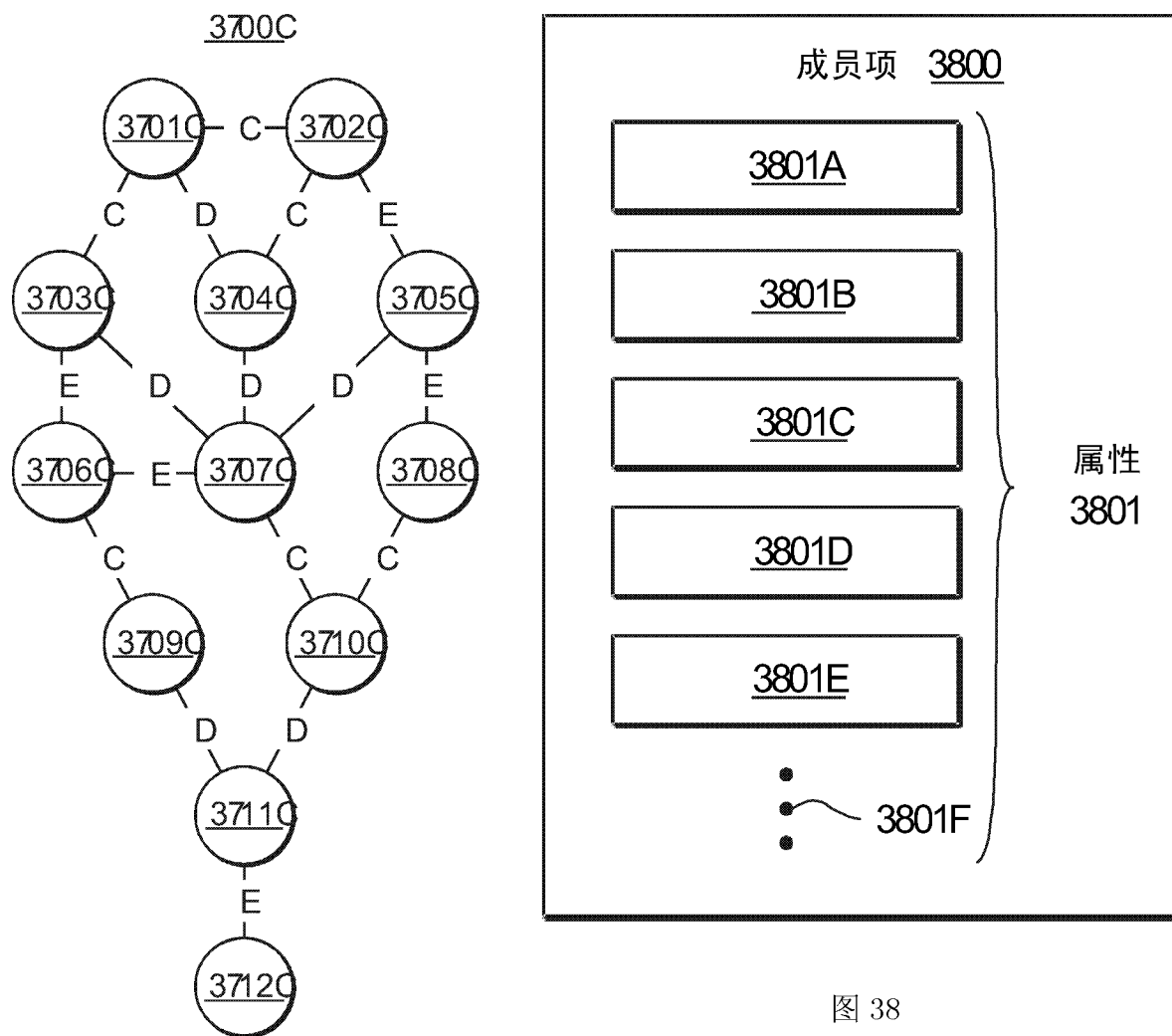


图 37C

图 38

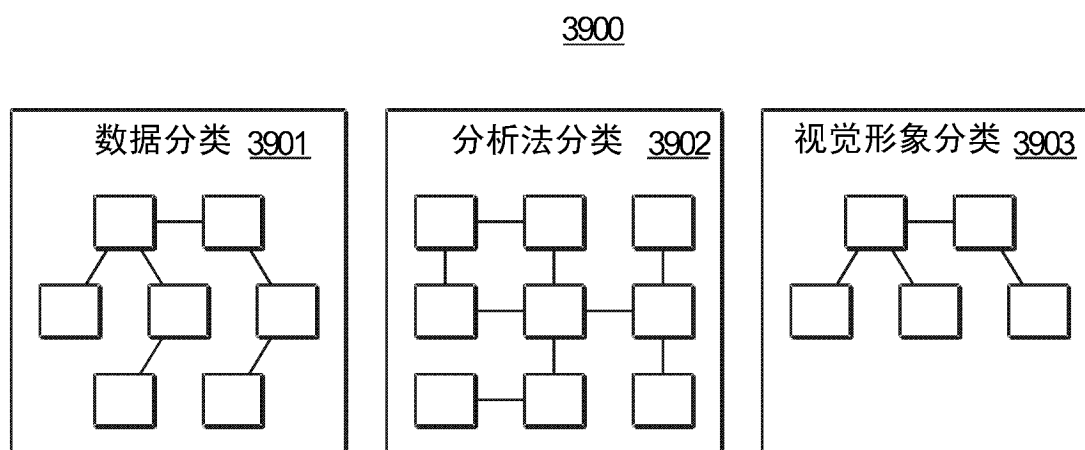


图 39

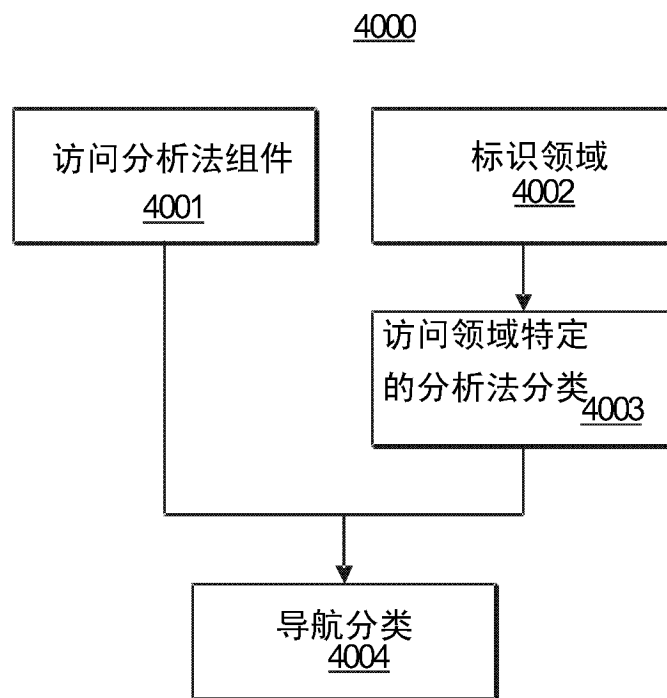


图 40

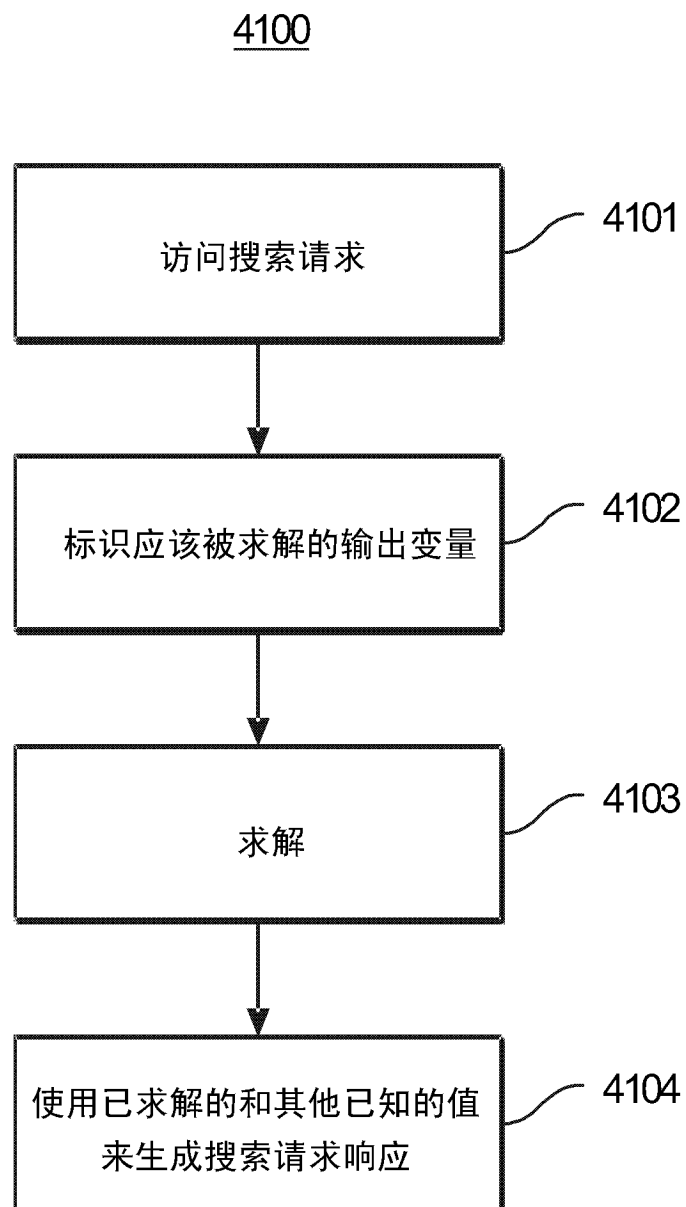


图 41

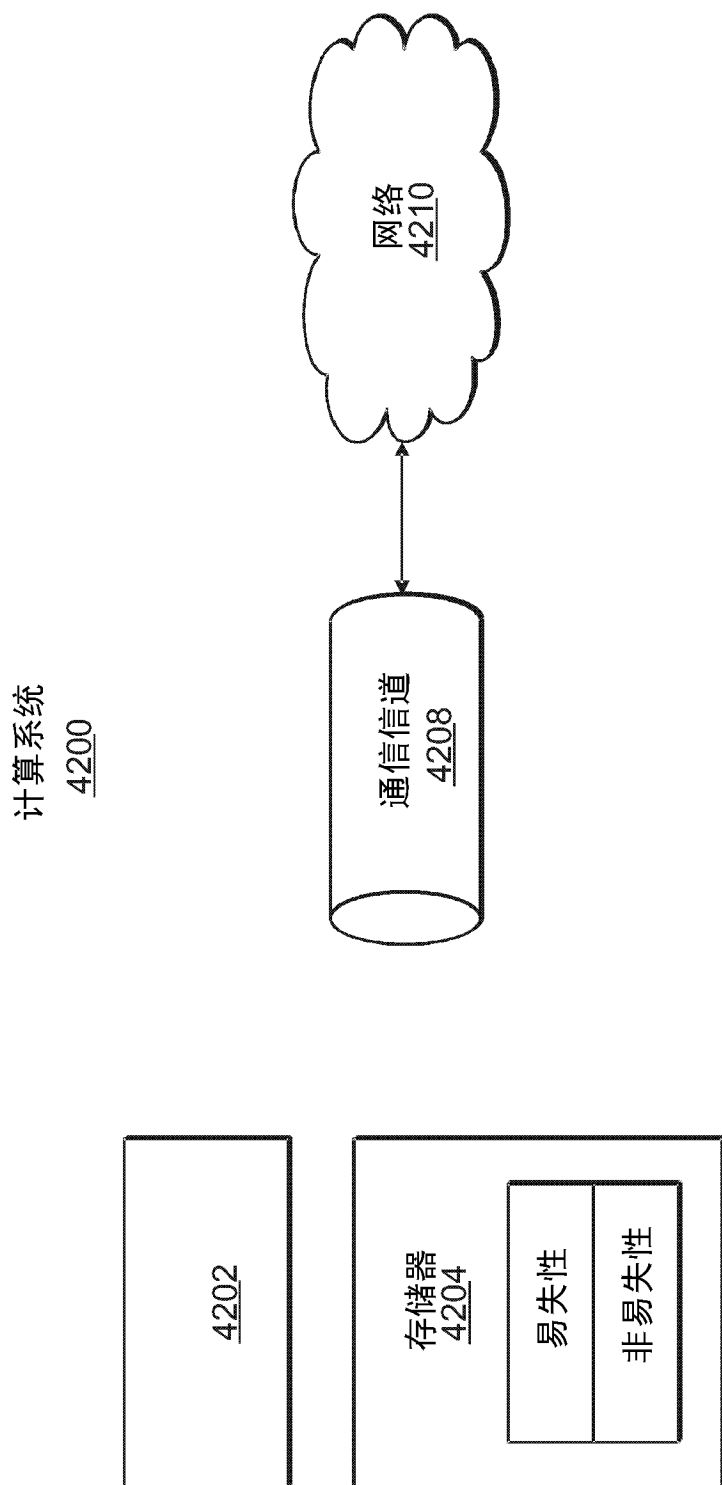


图 42