



(12) 发明专利申请

(10) 申请公布号 CN 118804823 A

(43) 申请公布日 2024. 10. 18

(21) 申请号 202380024678.0

(22) 申请日 2023.02.14

(30) 优先权数据

102022202145.2 2022.03.02 DE

(85) PCT国际申请进入国家阶段日

2024.08.30

(86) PCT国际申请的申请数据

PCT/EP2023/053646 2023.02.14

(87) PCT国际申请的公布数据

WO2023/165807 EN 2023.09.07

(71) 申请人 罗伯特·博世有限公司

地址 德国斯图加特

(72) 发明人 O·斯佩克托 D·迪卡斯特罗

V·楚耶夫

(74) 专利代理机构 中国专利代理(香港)有限公司 72001

专利代理师 才卓 刘春元

(51) Int.Cl.

B25J 9/16 (2006.01)

G06V 10/82 (2006.01)

G06T 7/73 (2006.01)

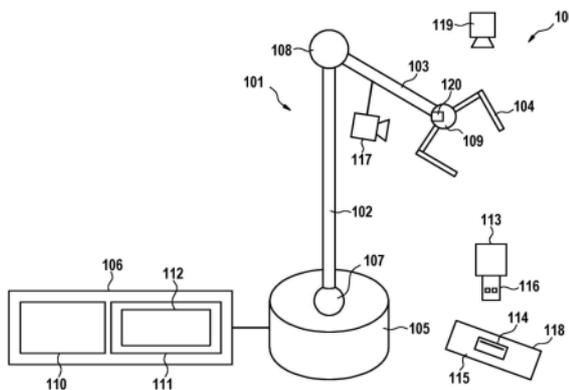
权利要求书1页 说明书10页 附图7页

(54) 发明名称

机器人和用于控制机器人的方法

(57) 摘要

根据各种实施例,描述了一种机器人,包括:末端执行器,具有带有至少第一指状物和第二指状物的夹持器,其中所述两个指状物被关于彼此布置成使得它们在它们之间限定夹持器平面;至少第一相机和第二相机,在夹持器平面的相对侧处被附接到末端执行器;以及控制器,被配置为针对末端执行器的定位,从第一相机接收第一图像并从第二相机接收第二图像;由神经网络处理第一图像和第二图像,其中所述神经网络被配置为输出用于插入任务的末端执行器的移动矢量;以及控制机器人如由移动矢量指定的那样移动。



1. 一种机器人,包括:
末端执行器,具有带有至少第一指状物和第二指状物的夹持器,其中所述两个指状物被关于彼此布置,使得它们在它们之间限定夹持器平面;
至少第一相机和第二相机,在夹持器平面的相对侧处被附接到末端执行器;
控制器,被配置为
针对末端执行器的定位,从第一相机接收第一图像并从第二相机接收第二图像;
由神经网络处理第一图像和第二图像,其中所述神经网络被配置为输出用于插入任务的末端执行器的移动矢量;以及
控制机器人如由移动矢量指定的那样移动。
2. 根据权利要求1所述的机器人,其中所述第一相机和所述第二相机被关于夹持器平面彼此对称布置。
3. 根据权利要求1或2所述的机器人,其中第一相机和第二相机的定位关于夹持器平面旋转。
4. 根据权利要求3所述的机器人,其中第一相机和第二相机的定位关于夹持器平面在30度和60度之间旋转,优选地在40度和50度之间旋转。
5. 根据权利要求1至4中任一项所述的机器人,其中所述控制器被配置为通过生成用于神经网络的输入图像并且将所述输入图像供应给神经网络来由神经网络处理第一图像和第二图像,所述输入图像具有与保持第一图像的图像数据的第一图像的通道数量相等的第一数量的通道,并且具有与保持第一图像的图像数据的第二图像的通道数量相等的第二数量的通道。
6. 一种用于控制机器人的方法,包括:
针对机器人的末端执行器的定位,从第一相机接收第一图像并从第二相机接收第二图像,其中所述末端执行器具有带有至少第一指状物和第二指状物的夹持器,其中所述两个指状物被关于彼此相对布置,使得它们在它们之间限定夹持器平面,并且其中所述第一相机和所述第二相机在夹持器平面的相对侧处被附接到末端执行器;
由神经网络处理第一图像和第二图像,所述神经网络被配置为输出用于插入任务的末端执行器的移动矢量;以及
控制机器人如由移动矢量指定的那样移动。
7. 根据权利要求6所述的方法,包括训练神经网络,以从输入数据元素中导出用于插入任务的移动矢量,所述输入数据元素包括从两个相机拍摄的图像数据。
8. 一种包括指令的计算机程序,当由处理器执行时,所述指令使处理器执行根据权利要求6至7中任一项的方法。
9. 一种存储指令的计算机可读介质,当由处理器执行时,所述指令使处理器执行根据权利要求6至7中任一项的方法。

机器人和用于控制机器人的方法

技术领域

[0001] 本公开涉及机器人和用于控制机器人的方法。

背景技术

[0002] 组装(诸如电气布线组装)是工业中最常见的体力劳动工作之一。示例是配电板组装和室内开关设备组装。复杂的组装过程典型地可以被描述为两个主要活动的序列:抓取和插入。类似的任务发生在例如线缆制造中,线缆制造典型地包括用于确认和验证的线缆插入。

[0003] 虽然对于抓取任务而言,合适的机器人控制方案在工业中典型地是可用的,但是由机器人执行插入或“轴孔装配”任务典型地仍然仅适用于小的问题子集,主要是牵涉固定位置中的简单形状的问题,并且其中不考虑变化。此外,现有的视觉技术很慢,典型地比人类操作员更慢三倍。

[0004] 因此,合期望的是用于训练针对机器人的控制器以执行像插入任务这样的任务的高效方法。

发明内容

[0005] 根据各种实施例,提供了一种机器人,所述机器人包括末端执行器,所述末端执行器具有带有至少第一指状物和第二指状物的夹持器,其中所述两个指状物被关于彼此相对布置,使得它们在它们之间限定夹持器平面,在夹持器平面的相对侧处附接到末端执行器的至少第一相机和第二相机,以及控制器,所述控制器被配置为针对末端执行器的定位,从第一相机接收第一图像并从第二相机接收第二图像,以通过神经网络处理第一图像和第二图像,其中所述神经网络被配置为输出用于插入任务的末端执行器的移动矢量,并且控制机器人如由移动矢量指定的那样移动。

[0006] 基于从布置在夹持器平面的相对侧处的相机拍摄的两个图像来确定移动矢量确保了神经网络具有充足的信息来导出用于插入任务的移动矢量。例如,即使一个指状物阻挡插入的视图,对于相机之一而言,另一个相机也将很可能具有插入的无阻挡视图。应当注意的是,对于插入任务而言,特别是良好的视图是重要的。更进一步地,具有两个相机允许神经网络导出深度信息。

[0007] 因此,两个相机(例如,关于夹持器平面对称布置)允许绕过一个图像模糊问题并且提取深度信息,同时在整个插入轨迹期间避免遮挡(如果一个相机的视图被遮挡,则另一个具有清晰的视图)。例如,每个相机被关于其相应的指状物开口成45度角放置,从而产生场景以及指状物之间的对象的良好视图。

[0008] 夹持器平面可以被理解为平坦对象在由夹持器夹持时被定向在其中的平面。

[0009] 下文给出了各种示例。

[0010] 示例1是如上面所描述的机器人。

[0011] 示例2是示例1的机器人,其中第一相机和第二相机被关于夹持器平面彼此对称地

布置。

[0012] 这增加了相机中的至少一个具有插入的无阻挡视图的机会。

[0013] 示例3是示例1或2的机器人,其中第一相机和第二相机的定位关于夹持器平面旋转。

[0014] 这进一步增加了相机中的至少一个具有插入的无阻挡视图的机会。

[0015] 示例4是示例3的机器人,其中第一相机和第二相机的定位关于夹持器平面在30度和60度之间旋转,优选地在40度和50度之间旋转。

[0016] 在典型情况下,该范围中的角度提供了相关部分(要被插入的对象的尖端,插入)的良好视图。

[0017] 示例5是示例1至4中任一个的机器人,其中控制器被配置为通过生成神经网络的输入图像并且向神经网络供应输入图像来由神经网络处理第一图像和第二图像,所述输入图像具有与保持第一图像的图像数据的第一图像的通道数量相等的第一数量的通道,并且具有与保持第一图像的图像数据的第二图像的通道数量相等的第二数量的通道。

[0018] 因此,这两个图像的图像数据被组合在单个图像中,并且神经网络可以以单个图像的方式对其进行处理(具有增加数量的通道,例如,用于两个RGB通道集的六个通道)。神经网络例如可以是或包括卷积神经网络。

[0019] 示例6是用于控制机器人的方法,包括针对机器人的末端执行器的定位,从第一相机接收第一图像并从第二相机接收第二图像,其中末端执行器具有带有至少第一指状物和第二指状物的夹持器,其中所述两个指状物关于彼此相对布置,使得它们在它们之间限定夹持器平面,并且其中第一相机和第二相机在夹持器平面的相对侧处被附接到末端执行器,由神经网络处理第一图像和第二图像,所述神经网络被配置为输出用于插入任务的末端执行器的移动矢量,并且控制机器人如由移动矢量指定的那样移动。

[0020] 示例7是示例6的方法,包括训练神经网络以从输入数据元素中导出插入任务的移动矢量,所述输入数据元素包括从两个相机拍摄的图像数据。

[0021] 示例8是包括指令的计算机程序,当由处理器执行时,所述指令使处理器执行根据示例6至7中任一个的方法。

[0022] 示例9是存储指令的计算机可读介质,当由处理器执行时,所述指令使处理器执行根据示例6至7中任一个的方法。

[0023] 应当注意,在机器人的上下文中描述的实施例和示例对于用于控制机器人的方法类似地有效,并且反之亦然。

附图说明

[0024] 在附图中,贯穿于不同的视图,相似的附图标记一般指代相同的部件。附图不一定按比例,代替地,重点一般放在图示本发明的原理上。

[0025] 在以下描述中,参考以下附图描述了各方面,其中:

[0026] 图1示出了机器人。

[0027] 图2更详细地示出了机器人末端执行器。

[0028] 图3图示了根据实施例的编码器网络的训练。

[0029] 图4示出了从图像数据元素和力输入中确定增量(delta)移动。

[0030] 图5示出了从两个图像数据元素中确定增量移动。

[0031] 图6图示了多步插入任务的示例。

[0032] 图7示出了图示用于控制机器人的方法的流程图。

[0033] 以下详细描述参考随附图,所述随附图通过图示的方式示出了可以在其中实践本发明的本公开的特定细节和各方面。在不脱离本发明的范围的情况下,可以利用其他方面,并且可以进行结构、逻辑和电气改变。本公开的各个方面不一定是互斥的,因为本公开的一些方面可以与本公开的一个或多个其他方面相组合以形成新的方面。

具体实施方式

[0034] 在下文中,将更详细地描述各种示例。

[0035] 图1示出了机器人100。

[0036] 机器人100包括机器人臂101,例如用于处置或组装工件(或者一个或多个其他对象)的工业机器人臂。机器人臂101包括操纵器102、103、104和通过其支撑操纵器102、103、104的基座(或支撑件)105。术语“操纵器”指代机器人臂101的可移动构件,对其的致动使得能够与环境进行物理交互,以例如实行任务。为了控制,机器人100包括(机器人)控制器106,(机器人)控制器106被配置为根据控制程序实现与环境的交互。操纵器102、103、104的最后一个构件104(距支撑件105最远)也被称为末端执行器104,并且可以包括一个或多个工具,诸如焊炬、夹持仪器、涂装装备等。

[0037] 其他操纵器102、103(更靠近于支撑件105)可以形成定位设备,使得与末端执行器104一起,提供在其末端处具有末端执行器104的机器人臂101。机器人臂101是能够提供与人类手臂类似功能的机械臂(可能在其末端处具有工具)。

[0038] 机器人臂101可以包括将操纵器102、103、104彼此互连并且与支撑件105互连的关节元件107、108、109。关节元件107、108、109可以具有一个或多个关节,所述关节中的每一个可以向相关联的操纵器提供相对于彼此的可旋转运动(即旋转运动)和/或平移运动(即位移)。操纵器102、103、104的移动可以借助于由控制器106控制的致动器来发起。

[0039] 术语“致动器”可以理解为被适配为响应于被驱动而影响机构或过程的组件。致动器可以将由控制器106发出的指令(所谓的激活)实现为机械移动。致动器(例如机电转换器)可以被配置为响应于驱动将电能转换成机械能。

[0040] 术语“控制器”可以被理解为任何类型的逻辑实现实体,其可以包括例如能够执行存储在存储介质中的软件的电路和/或处理器、固件或其组合,并且其可以发出指令,例如在本示例中向致动器发出指令。控制器可以例如通过程序代码(例如,软件)来配置,以控制系统的操作,在本示例中是机器人。

[0041] 在本示例中,控制器106包括一个或多个处理器110以及存储代码和数据的存储器111,处理器110根据所述代码和数据来控制机器人臂101。根据各种实施例,控制器106基于存储在存储器111中的机器学习模型112来控制机器人臂101。

[0042] 根据各种实施例,机器学习模型112被配置和训练为允许机器人100执行插入(例如,轴孔装配)任务,例如将插头113插入对应的插座114中。为此,控制器106借助于相机117、119拍摄插头113和插座114的照片。插头113例如是USB(通用串行总线)插头,或者也可以是电力插头。应当注意的是,如果插头像电力插头一样具有多个插栓(peg),那么每个插

栓均可以被视为要被插入的对象(其中所述插入物是对应的孔)。替代地,整个插头可以被看作要被插入的对象(其中所述插入物是电力插座)。应当注意的是(取决于什么被视为对象),对象113不一定被完全插入在插入物中。像在USB插头的情况下,如果金属接触部分116被插入插座114中,则认为USB插头被插入。

[0043] 用以执行轴孔装配任务的机器人控制典型地牵涉两个主要阶段:搜索和插入。在搜索期间,插座114被标识和定位,以提供用于插入插头113所要求的基本信息。

[0044] 搜索插入可以基于视觉或者基于牵涉例如螺旋路径的盲目策略。视觉技术很大程度上取决于相机117、119和板118(其中插座114被放置在板的表面115中)以及障碍物的位置,并且典型地比人类操作员更慢大约三倍。由于视觉方法的限制,控制器106可以考虑力-扭矩和触觉反馈,排他地或者与视觉相组合地考虑。

[0045] 在制造、检查和家庭服务机器人的设计中,构造可靠地插入不同对象(例如,插头、引擎齿轮)的机器人是一项巨大的挑战。最小化动作时间、最大化可靠性以及最小化抓取的对象和目标组件之间的接触由于涉及感测、控制的固有不确定性、对所施加的力的灵敏度以及遮挡而是困难的。

[0046] 根据各种实施例,提供了一种用以获取机器人策略的数据高效、安全并且有监督的方法。通过利用对比方法论和一次性学习技术,它允许利用很少的数据点学习控制策略,特别是用于多步插入任务的控制策略。

[0047] 根据各种实施例,训练和/或机器人控制包括如下中的一个或多个:

[0048] 1) 使用两个相机,以便避免一个图像模糊问题并且提取深度信息。这特别允许消除在插入对象113时接触插座的表面115的要求。

[0049] 2) 集成对比学习以便减少标记数据的量。

[0050] 3) 使得能够实现一次性学习和多步插入的关系网络。

[0051] 4) 使用该关系网络的多步插入。

[0052] 图2更详细地示出了机器人末端执行器201。

[0053] 末端执行器201例如对应于机器人臂101的末端执行器105,例如机器人臂六个自由度(DoF)。根据各种实施例,机器人具有两个感觉输入,控制器106可以使用这两个感觉输入以用于控制机器人臂101。第一个是立体感知,由两个(见上面的项目1)腕部相机202提供,所述两个腕部相机202以例如45°角倾斜,并且聚焦在末端执行器指状物(EFF) 203之间的点上。

[0054] 图像205、206是分别由第一相机和第二相机针对末端执行器201的定位拍摄的图像的示例。

[0055] 一般而言,每个相机202被定向成使得例如所拍摄的照片示出由末端执行器201夹持的要被插入的对象204的一部分(这里是销207)及其周围的区域,使得例如插入物208是可见的。应当注意,这里,插入物指代针对销的孔,但是它也可以指代包括针对销的孔以及针对插头的圆柱形部分的开口。因此,将对象插入到插入物中并不一定意味着对象被完全插入到插入物中,而是仅一个或多个部分被插入到插入物中。

[0056] 假设每个相机图像的高度 H 、宽度 W 和三个通道,机器人提供的臂的当前(或原点)定位的图像数据元素由 $\text{Img} \in \mathbb{R}^{H \times W \times 6}$ 标示(具有六个通道,因为它包括来自这两个

相机202的图像)。第二感觉输入是力输入,即力传感器120的测量,所述力传感器120测量当末端执行器105、201在平面(例如,板118的表面115)上按压对象113时由末端执行器105、201经历的力矩和力。力测量可以由机器人或由外力和扭矩传感器进行。力输入例如包括力指示 $F \triangleq (f_x, f_y, f_z) \in \mathbb{R}^3$ 和力矩指示 $M = (m_x, m_y, m_z) \in \mathbb{R}^3$ 。

[0057] 为了以下解释,机器人对当前定位的观察由 $\text{Obs} = (\text{Img}; F; M)$ 标示。为了准确地捕捉接触力并且生成平滑的移动,可以在传感器设备(相机202和力传感器120)和控制器106之间使用高频通信(实时数据交换)。

[0058] 例如,以500Hz采样力和扭矩(力矩)测量,并且利用125Hz在致动器处发送命令。

[0059] 末端执行器指状物203形成夹持器,其姿态由 L 标示。具体地, $L \triangleq [L_{\text{pos}}, L_{\text{ang}}]$,

其中 $L_{\text{pos}} \triangleq (x_0, y_0, z_0)$ 是夹持器的位置,并且 $L_{\text{ang}} \triangleq (\theta_0^x, \theta_0^y, \theta_0^z)$ 是其姿态。在笛卡尔空间中机器人的动作由 $(\Delta x, \Delta y, \Delta z, \Delta \theta_x, \Delta \theta_y, \Delta \theta_z)$ 定义,其中 Δx 、 Δy 和 Δz 是笛卡尔空间中的EEF相对于当前位置所需的期望校正。该机器人动作指定机器人从当前(或原点)姿态(特别是当前定位)到目标姿态(特别是目标定位)的移动。

[0060] 双相机方案,即对于所考虑的每个机器人定位具有包括两个图像的图像数据元素,允许恢复图像中所示的两个点之间的距离,即允许避免当试图使用单个图像在没有深度信息的情况下恢复世界坐标中的两个点之间的距离时出现的视觉模糊问题。

[0061] 根据各种实施例,(例如,通过控制器106)使用反向学习,其中,不仅在接触表面115之后,而且沿着移动轨迹,使用两个相机202的图像来收集训练数据(具体是图像)。这意味着收集训练数据元素,控制器106将机器人臂放置在其最终(目标)定位 L_{final} 中,即当插头116被插入到插座114中时(或者类似地,对于机器学习模型112应当被针对其进行训练的任何中间目标)。(应当注意,使用相机202,可以在根据如下面所描述的各种实施例使用的该定位中收集目标图像数据元素)。然后,对于每个训练数据元素,从概率(例如,正态)分布中采样两个点:一个是 T_{high} ,其被定位在插座上方的随机位置中,并且第二个是 T_{low} ,其被随机地定位在插座的高度周围。

[0062] 针对该训练数据元素的校正由 $\Delta_{\text{correction}} \triangleq L_{\text{final}} - T_{\text{random}}$ 定义,其中 T_{random} 是高点或低点(即 T_{high} 或 T_{low})。由 D 标示的训练数据集是基于训练任务集 $\tau \triangleq \{\tau_i\}_{i=1}^{N_\tau}$ 构建的,其中对于每个任务 τ_i ,存在对应的 L_{final}^i 。对于每个任务,生成随机化的点集 $\{T_{\tau_i,j}\}_{j=1}^{N_{\tau_i}}$,从而产生任务的起始点和终点以及初始随机点 $D \triangleq \cup_{\tau_i \in \tau} \{D_{\tau_i,j}\}_{j=1}^{N_{\tau_i}}$ (其中 $j = \text{高、低}$)以及 $D_{\tau_i,j} \triangleq [L_{\text{final}}^i, T_{\tau_i,j}]$ 。对于每个 $D_{\tau_i,j}$,定义对应的校正 $\Delta_{\tau_i,j} \triangleq L_{\text{final}}^i - T_{\tau_i,j}$ 。算法1给出了任务 τ 的数据收集和一般反向学习的详细示例。

```

Input: Maximum Randomized Location
 $r_0 = 10[\text{mm}]$ ; Maximum Randomized Angle
 $\theta_0 = 10^\circ$ ; Maximum Height  $z_{\max} = 50[\text{mm}]$ ;
Force threshold  $F_{\text{th}}$ ; Momentum threshold
 $M_{\text{th}}$ ; Points record  $P_{\text{rc}}$ ; Reaching time =  $t_{\max}$ ,
number of data points  $N_r$ , Buffer of Data  $D$ 
1 ; Init: Insert plug into the hole, register final pose
 $L_{\text{final}}$ 
2 Set:  $(\overbrace{x_0, y_0, z_0}^{X_0}, \overbrace{\theta_{x_0}, \theta_{y_0}, \theta_{z_0}}^{\Theta_0}) = L_{\text{final}}$ 
3 for  $i \leftarrow 1, \dots, N_r$  do
4    $p_X^0, p_X^1 \sim \mathcal{N}(X_0, r_0)$ 
5    $p_\Theta^0, p_\Theta^1 \sim \mathcal{N}(\Theta_0, \theta_0)$ 
6    $z^1 \sim \text{Uniform}[z_0, z_{\max}]$  random height above  $z_0$ 
7   Set target  $T_{\text{high}}^i = (p_X^0, z_1, P_\Theta^0)$ 
8   Set target  $T_{\text{low}}^i = (p_X^1, z_0, P_\Theta^1)$ 
9   for  $j \leftarrow [\text{high}, \text{low}]$  do
10    Move Robot to  $T_j^i$  using an impedance
        controller
11    while Robot did not reach  $T_j^i$  do
12      if time mod  $t_{\max}/P_{\text{rc}}$  then
13        Register Pose  $E = \{x, y, z, \theta_x, \theta_y, \theta_z\}$ 
14        Capture two  $45^\circ$  cameras into  $\text{Img}$ 
15        Capture force  $F$  and momentum  $M$ 
16        Set  $\text{Obs} = \{\text{Img}, F, M\}$ 
17        Set  $\text{Action}_i = \{L_{\text{final}} - E\}$ 
18        Push  $\{\text{Obs}, \text{Action}\}$  into  $D$ 
19      end
20    end
21  end
22  return  $B$ 
23 end

```

[0064] 算法1

[0065] 根据算法1,收集力传感器数据。根据各种实施例,那是不必要的,特别是利用目标图像数据元素操作的那些,如下面参考图5进一步描述的那样。

[0066] 根据各种实施例,机器学习模型112包括多个组件,它们中的一个编码器网络,控制器106使用所述编码器网络来确定每个图像数据元素 Img 的编码。

[0067] 图3图示了根据实施例的编码器网络301的训练。

[0068] 编码器网络301例如是卷积神经网络,例如具有ResNet18架构。

[0069] 编码器网络301(实现函数 Φ)使用对比损失,以及增量策略损失和关系数据损失中的一个或这两者来训练,因此例如根据以下损失:

[0070] 损失 = $l_{\text{constrastive}} + l_{\text{delta}} + l_{\text{relation}}$ 。

[0071] 这些损失分量在下文中描述。

[0072] 对比学习,即基于对比损失的训练,是服从分别映射到正标签或负标签上的数据集中的相似性或相异性约束的学习表示的框架。一种可能的对比学习方法是实例辨别,其中如果示例和图像是同一实例的数据增强,则它们是正对,并且否则是负对。对比学习中的关键挑战是负样本的选择,因为它可能影响所学习的底层表示的质量。

[0073] 根据各种实施例,使用对比技术(例如,由控制器106或者由之后存储在控制器106

中的外部设备)来训练编码器网络301,使得它在没有任何特定标签的情况下学习手头上的任务的相关特征。示例是InfoNCE损失(NCE:噪声对比检查)。由此,通过将来自两个相机202的两个图像堆叠到图像数据元素302,获得插头113和插座114的深度配准。该深度信息被用于以各种不同的方式增强图像数据元素。从原始的(即非增强的图像数据元素)和以该方式获得的一个或多个增强中,对图像数据元素对进行采样,其中该对中的一个元素被供应给编码器网络301,并且另一个被供应给编码器网络的另一版本303。所述另一版本303实现函数 ϕ' ,函数 ϕ' 具有例如与编码器网络301相同的参数,并且使用根据 $\phi' = \phi' + (1 - m)\phi$ (其中 $m=0.999$)的Polyak平均来更新(其中 ϕ 、 ϕ' 已经被用于表示两个编码器网络版本301、302的权重)。

[0074] 两个编码器网络版本301、302每个输出输入数据元素302的大小为L的表示(即嵌入)(即该对的 $2 \times L$ 输出)。针对N个图像数据元素的批(即,针对每个图像数据元素形成增强对或原始和增强),即,针对大小为 $N \times 6 \times H \times W$ 的训练输入图像数据这样做,给出了由两个编码器网络版本301、302输出的N个表示对(即,大小为 $2 \times N \times L$ 的表示输出数据)。使用这些N个表示对,通过从被包括在所述对中的表示形成正对和负对来计算编码器网络301的对比损失。这里,如果两个表示是从相同的输入数据元素302生成的,则它们是正对,并且如果它们是从不同的输入数据元素302生成的,则它们是负对。这意味着正对保持同一原始图像数据元素的两个增强,或者保持原始图像数据元素及其增强。所有其他对都是负对。

[0075] 为了确定对比损失304,使用相似性函数 $\text{sim}(\cdot)$ (或者两个嵌入之间的距离)来度量相似性。它可以使用欧几里德距离(在潜在空间中,即嵌入的空间中),但是可以使用更复杂的函数,例如使用核。于是,对比损失例如由下式的 i, i 上的和给出:

$$[0076] \quad \ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \exp(\text{sim}(z_i, z_k)/\tau)}$$

[0077] 其中 z_i 是嵌入。 T 在这里是温度归一化因子(不要与上面使用的任务 τ 相混淆)。

[0078] 图4示出了从图像数据元素401和力输入402中确定增量移动405。

[0079] 增量移动 $\Delta = (\Delta x, \Delta y, 0, \Delta \theta_x, \Delta \theta_y, \Delta \theta_z)$ 是在 z 方向上没有移动的机器人动作 $\Delta = (\Delta x, \Delta y, \Delta z, \Delta \theta_x, \Delta \theta_y, \Delta \theta_z)$,因为根据各种实施例,控制器106独立于其他信息来控制 z 方向上的移动 Δz ,例如使用来自现有知识或来自深度相机的在其处放置插座114的桌子的高度。

[0080] 在该情况下,编码器网络403(对应于编码器网络301)为图像数据元素生成嵌入。嵌入与力输入402一起被传递到提供增量移动405的神经网络404。神经网络404例如是卷积神经网络,并且被称为增量网络,并且据说实现增量(控制)策略。

[0081] 通过具有用于训练输入数据元素(包括图像数据元素401和力输入402,即在上面特别是在算法1中由 $\text{Obs} = (\text{Img}; \text{F}; \text{M})$ 标示)的基准真值增量移动标签,确定用于训练编码器网络403的增量损失 l_{delta} 。由算法1生成的训练数据集 D 包括训练输入数据元素 Obs 和增量损失的基准真值标签。

[0082] 图5示出了从两个图像数据元素501、502中确定增量移动505。

[0083] 在该情况下,编码器网络503(对应于编码器网络301)针对每个图像数据元素501、502生成嵌入。嵌入被传递到提供增量移动505的神经网络504。神经网络504例如是卷积神

神经网络,并且被称为关系网络,并且据说实现关系(控制)策略。

[0084] 用于训练编码器网络503(以及关系网络504)的关系损失 l_{relation} 通过具有图像数据元素对501、502的基准真值增量移动标签来确定。图像数据元素对501、502的基准真值增量移动标签可以例如通过取由算法1生成的数据集D中的图像数据元素所包括的基准真值增量移动标签(即,动作)之间的差来生成。

[0085] 这意味着,对于具有关系损失的训练,数据集D被用于通过由差 $\Delta_{\tau_k,j} = \Delta_{\tau_i,j} - \Delta_{\tau_i,k}$ 计算基准真值来计算相同插接任务的两个图像 Img_i 和 Img_j 之间的增量移动,其中 $j \neq k$ 。在 Img_i 和 Img_j 针对该训练被增强的情况下,使用一致的增强。

[0086] 关系损失 l_{relation} 促进一次性学习以及使得能够实现多步插入,并且改进对所收集数据的利用。

[0087] 当被训练时,如参考图4所描述的用以从图像数据元素401和力输入402导出增量移动的编码器网络403和增量网络404,实现被称为增量(控制)策略的策略。类似地,如参考图5所描述的用以从图像数据元素501(针对当前定位)和图像数据元素502(针对目标定位)导出增量移动的经训练的编码器网络503和关系网络504实现被称为关系(控制)策略的策略。

[0088] 控制器可以使用增量策略或关系策略作为与主策略 π_{Main} 相组合的剩余策略 π_{Residual} 。因此,为了推断,控制器106将编码器网络403、503用于增量策略或关系策略。这可以取决于用例来决定。例如,对于一次性或多步插入任务,使用关系策略(和根据图5的关系架构),因为它可以在这些任务中更好地泛化。对于其他任务,使用增量策略(和图4的增量架构)。

[0089] 遵循主策略,控制器106近似孔的位置,即例如定位孔、插座、螺纹等。在场景中,例如从图像中,并且使用例如PD控制器来跟随从所述近似中计算的路径。

[0090] 然后,它激活剩余策略,例如在插头113距表面115的特定R处,并且根据剩余策略进行实际插入。剩余策略的动作是增量移动:

[0091] $\Delta = (\Delta x, \Delta y, 0, \Delta \theta_x, \Delta \theta_y, \Delta \theta_z)$ 。

[0092] 算法2给出了该程序的详细示例。

Input: Target Point T ; Transition Radius R ; Final position image Img_F ; Lowering rate Δ_Z ; Final height in z -axis z_{final} ; Choose either $\pi_{\text{Residual}} = \pi_{\text{Delta}}$ (Fig. 4) Or π_{Relation} (Fig. 5) according to task in hand.

- 1 **Init:** $\text{NextPoint} = T$, $\text{FinishFlag} = \text{False}$
- 2 **while** not FinishFlag **do**
- 3 Robot move to NextPoint
- 4 $\text{CurrentPoint} = \text{Measure}(x, y, z, \theta^x, \theta^y, \theta^z)$
- [0093] 5 **if** $\text{dist}(\text{CurrentPoint}, T) < R$ **then**
- 6 $\Delta = \pi_{\text{Residual}}(O, \text{Img}_F)$
- 7 $\delta_z = (0, 0, R/\Delta_Z, 0, 0, 0)$
- 8 $\text{NextPoint} = \text{CurrentPoint} + \Delta + \delta_z$
- 9 **if** $z < z_{\text{final}}$ **then**
- 10 $\text{FinishFlag} = \text{True}$
- 11 **end**
- 12 **end**
- 13 **end**

[0094] 算法2

[0095] 根据各种实施例,为了提高鲁棒性以及对于颜色和形状的泛化,可以使用用于训练数据元素的各种增强。次序以及每个增强的属性对泛化具有很大的影响。针对视觉增强(即,训练图像数据的增强),对于基于增量损失和关系损失的训练,这可以包括重新定大小、随机裁剪、颜色抖动、平移、旋转、擦除和随机卷积。对于对比损失,示例性增强是随机重新定大小的裁剪、强平移、强旋转和擦除。类似的增强可以被用于训练同一批内的数据元素。关于力增强(即训练力输入数据的增强),矢量(F,M)的方向,而不是它们的幅度,典型地是更重要有价值的因子。因此,根据各种实施例,至增量网络的力输入402是力和力矩矢量的方向。这些可以被增强以用于训练(例如,通过抖动)。

[0096] 如上面所提及的,关系策略特别是可以被用于多步任务,例如多步插入。应当注意,在多步插入任务(比如锁门)中,典型地更难以收集训练数据并且验证每个步骤都可以被完成。

[0097] 根据各种实施例,对于多步任务,针对包括一个或多个中间目标和最终目标的每个目标预先保存图像。然后,对于当前要实现的每个目标(取决于当前步骤,即,根据中间目标的序列,以及作为最后元素的最终目标),取得针对当前定位的图像数据元素(例如,通过从这两个相机202取得图像),并且与目标的(一个或多个)图像一起馈送到编码器网络503,并且如参考图5所描述的那样由关系网络504导出增量移动。

[0098] 图6图示了多步插入任务的示例。

[0099] 在图6的示例中,用以锁门的任务,该任务由三个步骤构成,所述三个步骤从初始定位601开始:插入钥匙602,转动锁603,并且然后转回来604。对于每个步骤,状态的图像被利用两个(例如,45度)相机202来拍摄并且预先保存。任务的执行遵循具有关系策略和相似性函数的算法2,以在步骤之间切换。例如,当所拍摄的图像和当前步骤的预先保存的目标

图像之间的相似性低于阈值时,认为达到了中间目标。

[0100] 即使使用了常规的反向数据收集(例如根据算法1),在训练中也不访问锁定和插入状态本身(仅访问孔表面上方或接触孔表面的状态),控制器106可以以该方式成功地执行任务。

[0101] 总之,根据各种实施例,提供了如图7中所图示的方法。

[0102] 图7示出了图示用于控制机器人的方法的流程图700。

[0103] 在701中,针对机器人的末端执行器的定位,从第一相机接收第一图像并从第二相机接收第二图像,其中末端执行器具有带有至少第一指状物和第二指状物的夹持器,其中所述两个指状物被关于彼此布置,使得它们在它们之间限定夹持器平面,并且其中第一相机和第二相机在夹持器平面的相对侧处被附接到末端执行器。

[0104] 在702中,第一图像和第二图像由神经网络处理,所述神经网络被配置(即,特别被训练)为输出用于插入任务的末端执行器的移动矢量。

[0105] 在703中,控制机器人如由移动矢量指定的那样移动。

[0106] 图7的方法可以由包括一个或多个数据处理单元的一个或多个计算机来执行。术语“数据处理单元”可以被理解为允许处理数据或信号的任何类型的实体。例如,可以根据由数据处理单元执行的至少一个(即,一个或多于一个)特定功能来处理数据或信号。数据处理单元可以包括模拟电路、数字电路、复合信号电路、逻辑电路、微处理器、微控制器、中央处理单元(CPU)、图形处理单元(GPU)、数字信号处理器(DSP)、可编程门阵列(FPGA)集成电路或其任何组合,或者由其形成。实现相应功能的任何其他方式也可以被理解为数据处理单元或逻辑电路。将理解,本文中详细描述的方法步骤中的一个或多个可以由数据处理单元通过由数据处理单元执行的一个或多个特定功能来执行(例如,实现)。

[0107] 各种实施例可以接收和使用来自诸如视频、雷达、LiDAR、超声波、热成像等各种视觉传感器(相机)的图像数据。实施例可以被用于训练机器学习系统和控制机器人,例如自主地控制机器人操纵器,以实现不同场景下的各种插入任务。应当注意,在针对插入任务进行训练之后,可以针对新的插入任务训练神经网络,与从头开始训练相比,这减少了训练时间(迁移学习能力)。实施例特别适用于控制和监视例如组装线中操纵任务的执行。

[0108] 根据一个实施例,所述方法是计算机实现的。

[0109] 尽管本文中已经图示和描述了特定实施例,但是本领域普通技术人员将领会,在不脱离本发明的范围的情况下,可以用各种各样的替代和/或等同实现来替换所示出和描述的特定实施例。本申请意图覆盖本文中讨论的特定实施例的任何适配或变化。因此,意图的是本发明仅受权利要求及其等同物的限制。

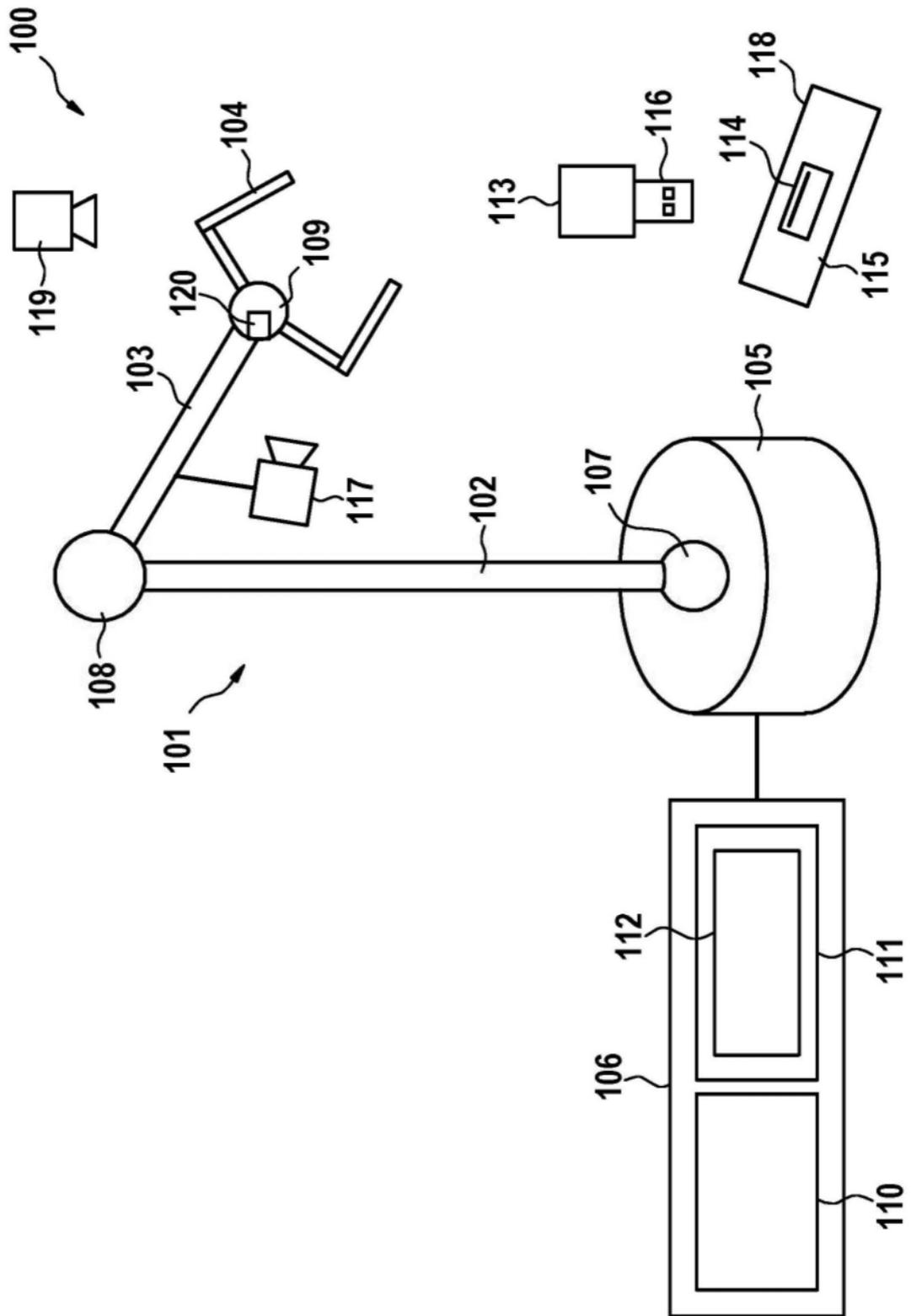


图1

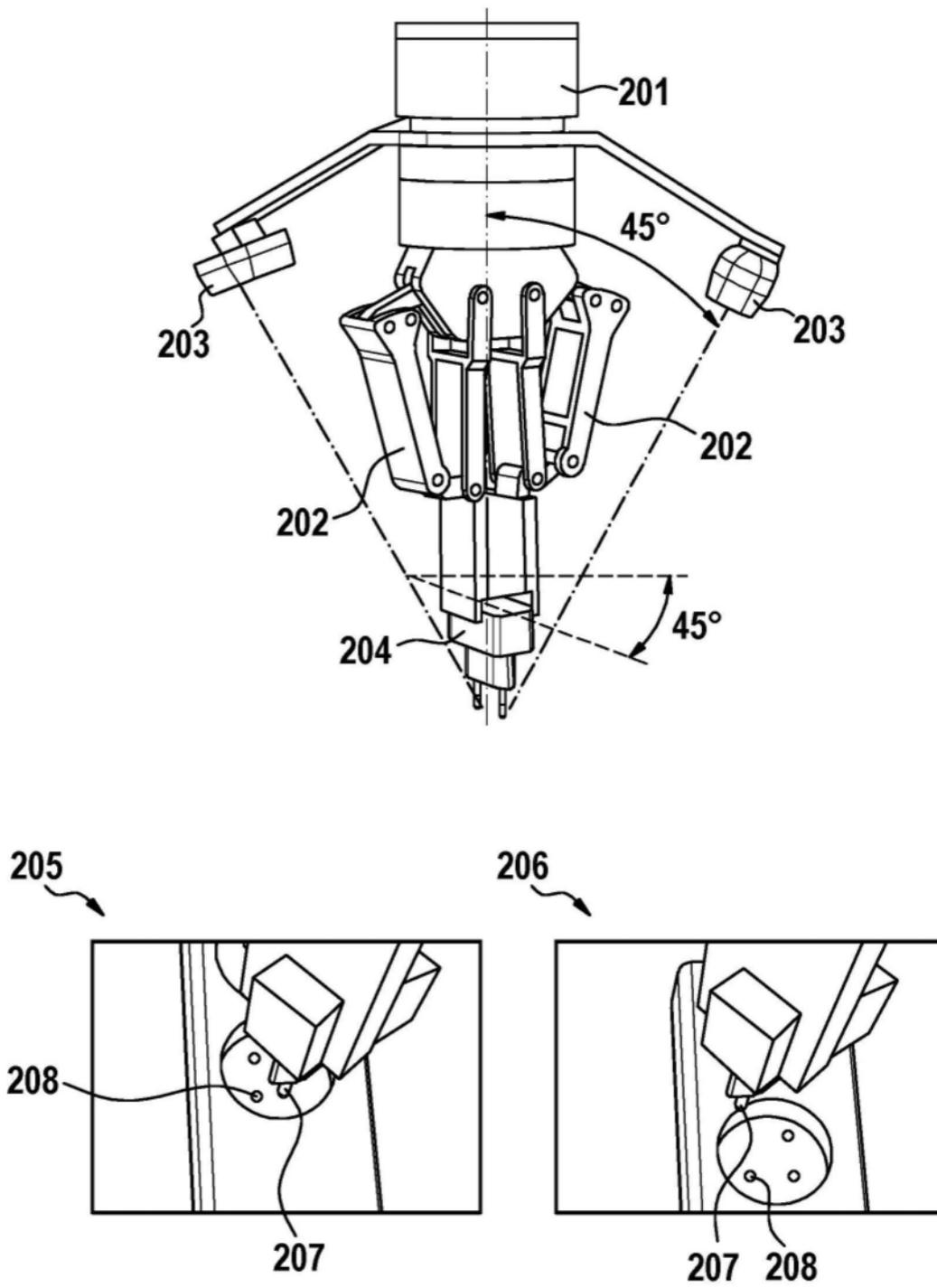


图2

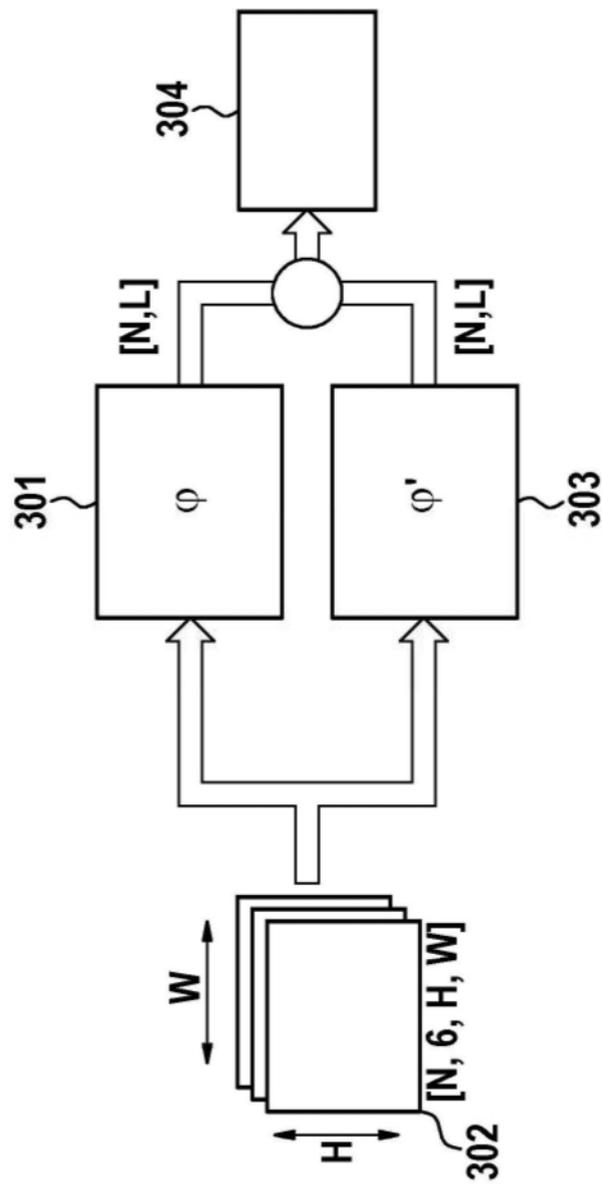


图3

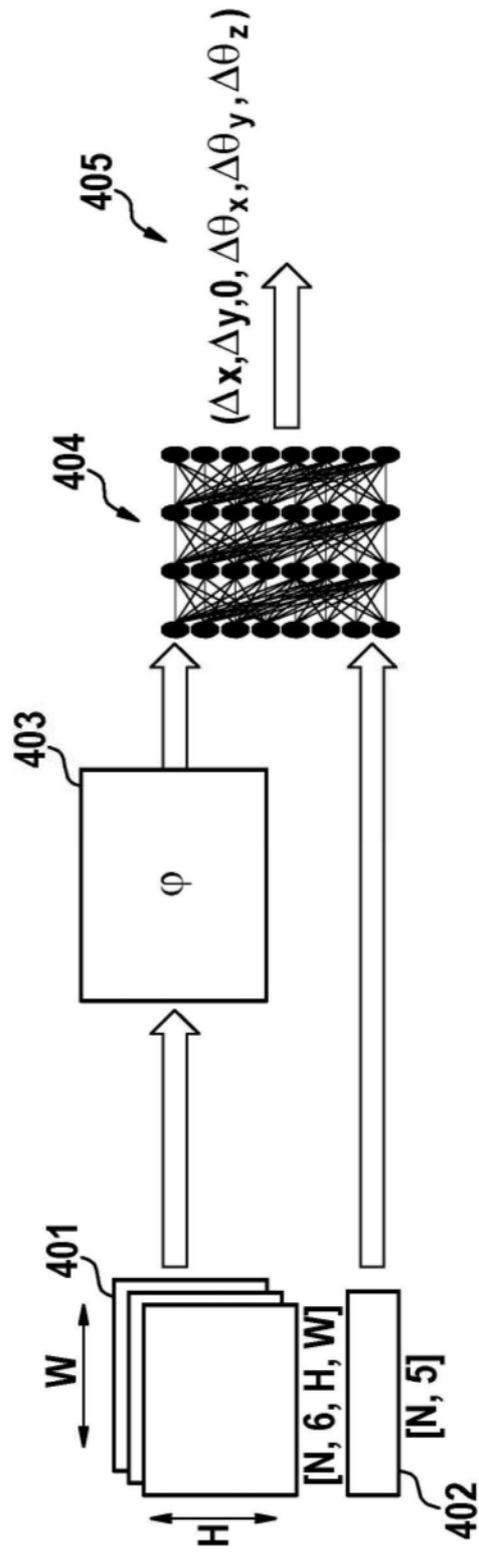


图4

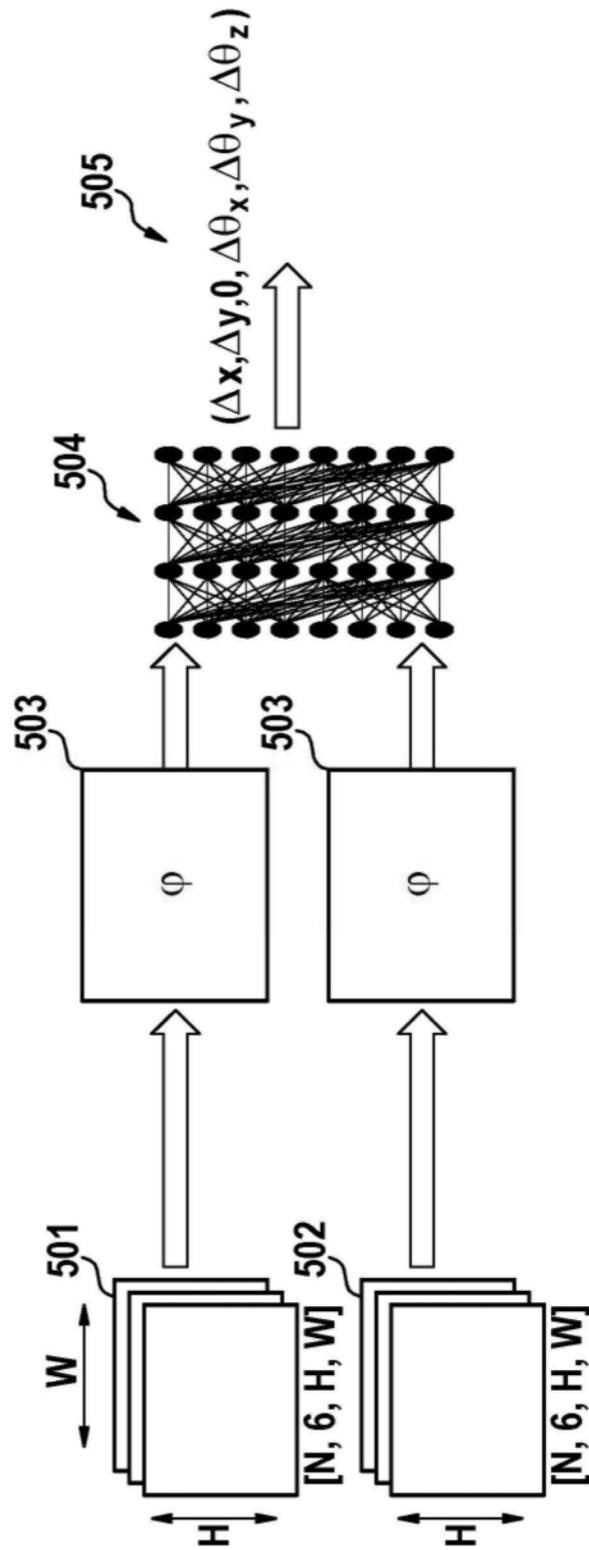


图5

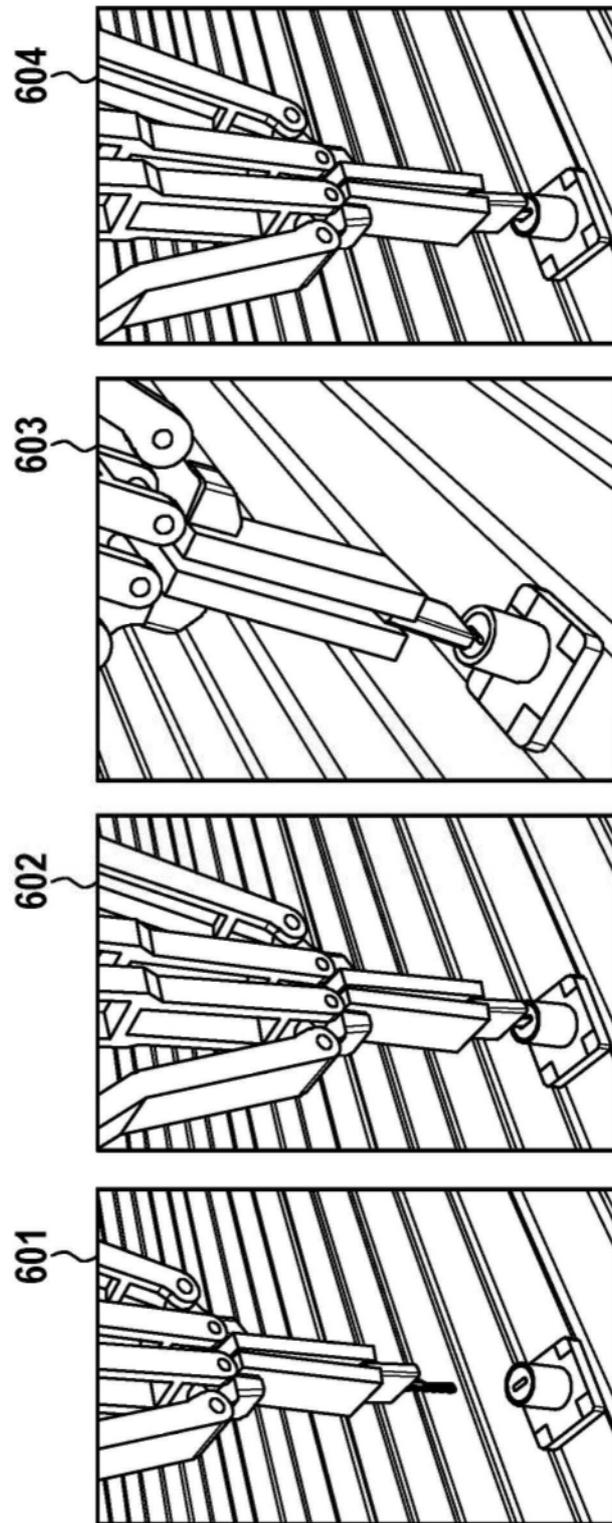


图6

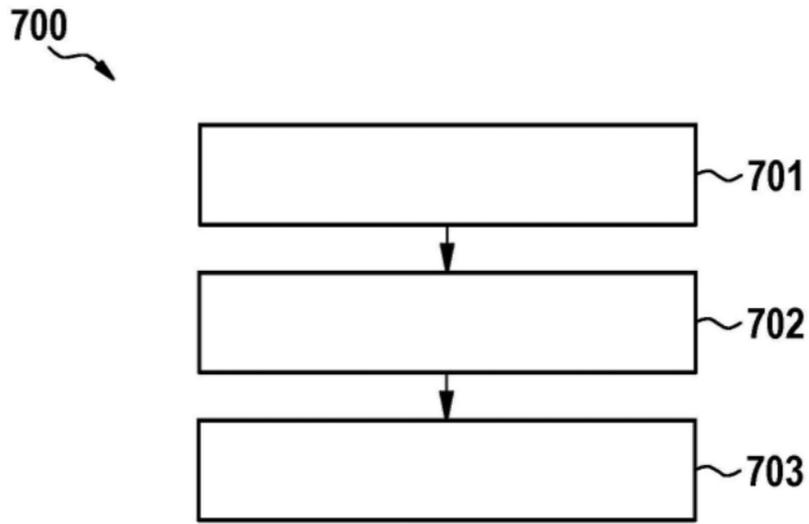


图7