



(12)发明专利申请

(10)申请公布号 CN 106826822 A

(43)申请公布日 2017.06.13

(21)申请号 201710056272.7

(22)申请日 2017.01.25

(71)申请人 南京阿凡达机器人科技有限公司

地址 211316 江苏省南京市高淳经济开发区古檀大道3号

(72)发明人 张光肖

(74)专利代理机构 江苏圣典律师事务所 32237

代理人 贺翔 刘辉

(51)Int.Cl.

B25J 9/16(2006.01)

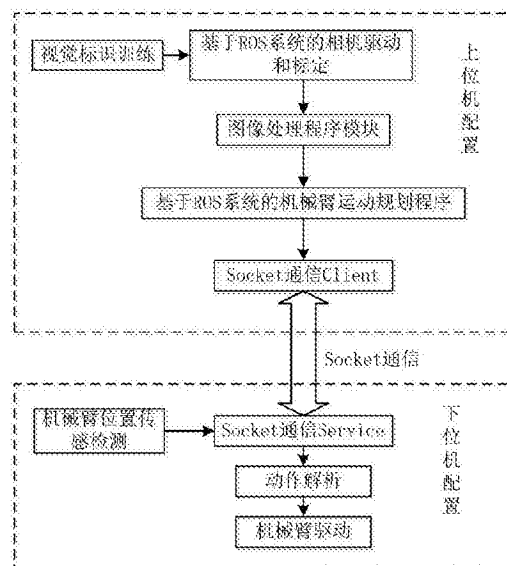
权利要求书2页 说明书6页 附图5页

(54)发明名称

一种基于ROS系统的视觉定位及机械臂抓取实现方法

(57)摘要

本发明提出一种基于ROS系统的视觉定位及机械臂抓取实现方法,实现过程如下:首先,相机置于待抓取物体的上方或侧上方并获取被抓取物体及附属标记的图像,输入到上位机;然后在上位机上配置相机驱动,读取相机的图像并利用特定的算法进行数据处理,获取物体相对于相机和机械臂的空间坐标并给出运动规划参数值发送给下位机;最后下位机接收并解析上位机发送的运动信息,驱动机械臂按照预定动作进行执行抓取。本发明提出的基于ROS系统的单目视觉定位及机械臂抓取实现方法可以有效地利用上位机强大的处理运算能力,又容易实现机械臂及上位机的布局和多机械臂协同工作,同时采用ROS系统可以轻松实现机械臂运动规划具有广泛的应用前景。



1. 一种基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征在于,包括以下步骤:

步骤1:安装相机,并将所述相机连接至上位机;

步骤2:配置所述相机在ROS系统下的使用环境;

步骤3:所述上位机在ROS系统下进行图像处理,包括:

步骤31:读取所述相机的矫正数据,并选择要采用的识别算法;

步骤32:定位标记识别,给出空间位置信息;

步骤33:将获得的空間位置信息传递给MoveIt模块进行运动规划;

步骤4:所述上位机在ROS系统下进行机械臂运动规划;

步骤5:将获得的机械臂运动信息队列传递至下位机;

步骤6:所述下位机接收机械臂运动信息队列并解析,同时驱动机械臂按照规划的路径执行运动和抓取,并将机械臂的实时位姿回传至所述上位机。

2. 根据权利要求1所述的基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征在于,所述步骤2中配置相机的使用环境包括:

步骤21:在ROS系统下配置USB_CAM对所述相机进行驱动,同时将所述相机采集的图像信息发布出去;

步骤22:在ROS系统下对所述相机进行标定并保存矫正数据;

步骤23:执行视觉标识训练。

3. 根据权利要求2所述的基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征在于,所述执行视觉标识训练的过程为:

对于基于ARToolKit的特定标记识别算法,利用在线工具“Tarotaro”训练或者使用ARToolKit提供的mk_patt的离线工具进行训练;

对于基于OpenCV_ArUco的识别算法,则利用drawMarker()函数进行创建和训练标识图案。

4. 根据权利要求2所述的基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征在于,所述步骤32选择的识别算法为基于ARToolKit的识别算法或基于OpenCV_ArUco的识别算法:

如果选择基于ARToolKit的识别算法,程序处理过程为:读取预先导入的标识图信息,所述相机获取实时的图像,寻找可靠度最高的标识并定位相机观察到的标识,在标识上放置系统原点,添加ARToolKit 帧频计数器,利用不同帧下标识图的位置变化预估标识图空间中的位置并实时发布,同时利用OpenGL在标识图标上绘制坐标系;

如果选用基于OpenCV_ArUco的识别算法,首先,获取图像中所有的可疑标识,并依次提取标识图像中的标识位;

其次,利用透视变换获取标准标识图,然后利用OTSU二值化算法读取标识位的值,并将所述标识位的值与预先训练的标识图像作比较来实现标识图识别;

再次,对已识别的标识图进行空间位姿估计,并给出已识别的标识图的空间位置信息;

最后,根据预先设定的相机坐标系和机械臂坐标系转换矩阵,获取带有标识图的物体相对于机械臂的空间位置信息,即所述空间位置信息。

5. 根据权利要求2所述的基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征

在于,所述步骤32定位标记识别,给出空间位置信息的过程包括:

步骤321:获取图像中所有的可疑标识,并依次提取标识图像中的标识位;

步骤322:利用透视变换获取标准标识图,然后利用OTSU二值化算法读取出标识位的值,并将所述标识位的值与预先训练的标识图像作比较来实现标识图识别;

步骤323:对已识别的标识图进行空间位姿估计,并给出已识别的标识图的空间位置信息;

步骤334:根据预先设定的相机坐标系和机械臂坐标系转换矩阵,获取带有标识图的物体相对于机械臂的空间位置信息,即所述空间位置信息。

6.根据权利要求1所述的基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征在于,所述步骤4上位机在ROS系统下进行机械臂运动规划的过程为:

步骤41:采用urdf编写机械臂模型描述文件,用于机械臂的建模;

步骤42:采用ROS系统下的MoveIt 初始化工具包调用所创建的机械臂描述模型;

步骤43:机械臂建模;

步骤44:读取图像处理获得的所述空间位置信息,利用MoveIt模块进行路径规划,然后将规划的机械臂运动信息队列发布出去。

7.根据权利要求1所述的基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征在于,所述上位机和下位机之间的通信,采用Socket通信方式。

8.根据权利要求6所述的基于ROS系统的单目视觉定位及机械臂抓取实现方法,其特征在于,所述通信过程为:

步骤71:消息服务器模块初始化,用于读取MoveIt模块发布的运动规划信息队列;

步骤72:初始化Socket通信信息,并将信息队列放置在发送缓冲区,在上下位机进行通信时发送给下位机;

步骤73:下位机接收到数据后进行运动信息解析,驱动机械臂按照规划动作执行抓取;

步骤74:在执行规划动作的同时,机械臂上的位置传感器将机械臂的实际位姿通过Socket发送给上位机。

一种基于ROS系统的视觉定位及机械臂抓取实现方法

[0001]

技术领域

[0002] 本发明属于机械臂控制和运动规划领域,特别是一种基于ROS系统的视觉定位及机械臂抓取实现方法。

背景技术

[0003] 机械臂是在机器人领域中应用最广泛的一种自动化装置,尤其是多自由度机械臂在机械制造、汽车、半导体、医疗、家庭服务等多个领域扮演越来越多的角色,因此机械臂的运动控制一直是研究的热点。目前,机械手臂最主要的应用场景主要有以下几个:

1)焊接领域。用于代替人工在不良的焊接环境中执行焊接人物。

[0004] 2)自动化生产线领域。主要用于执行物品的抓取、翻转、物品分拣等动作,提高生产效率。

[0005] 3)医疗领域。主要用于执行精密的医疗操作,如微创手术等。

[0006] 4)服务领域。配合移动机器人,机械臂已经走入生活之中执行如递取物品、收拾杂物等任务。

[0007] 由于条件限制,机械臂的智能性仍然不够高,更多的是执行机械的示教动作,对于多变的环境应用仍然存在诸多问题,本发明旨在提高机械臂的智能化和环境适应性。

[0008] 机器视觉属于人工智能的一个分支,简而言之,机器视觉就是用相机代替人眼来对周围环境进行判断和分析,结合一定的算法来实现智能决策,它是一项综合技术,包括图像处理、机械工程技术、控制、电光源照明、光学成像、传感器、模拟与数字视频技术、计算机软硬件技术等。机器视觉从原理上分为单目、双目、3D视觉等几个类型,它的引入存在以下几个优势:

1)机器视觉相对人眼具有更高得可靠性。机器视觉可以连续采集图像和连续工作,不会出现视觉疲劳等情况。

[0009] 2)机器视觉具有更高的精度。配合一定的处理算法,机器视觉可以实现精确测量和误差检验,且有利于数据记录和集成。

[0010] 3)机器视觉可以适应复杂环境。在一些不适合人工作业的场合,机器视觉则可以“大展身手”。

[0011] 将机器视觉和机械臂结合起来,相当于为机械臂增加了智能的“眼睛”,可以大大增加机械臂的环境感知能力和智能决策能力,从而进一步扩展机械臂的应用领域。本发明就是在此基础上提出了一种基于单目视觉的定位方法,配合机械臂可以轻松实现物品的抓取和放置等动作。

[0012] ROS系统(Robot Operating System)是Willow Garage公司2010年发布的开源机器人操作系统。它采用分布式的组织架构,可以大大提高代码的复用性和环节复杂机器人系统的适应性。ROS系统主要有以下几个特点:

1) 点对点的分布式设计。ROS的点对点设计以及服务和节点管理等机制可以分散由计算机视觉和语音识别等功能带来的实时计算压力,能够适应多机器人遇到的挑战。

[0013] 2) 多语言的支持。ROS系统支持C++、Python、Octave和LISP等编程语言,同时提供了其他编程语言的接口。

[0014] 3) 软件包丰富。ROS系统集成了大量的软件包,可以快速的实现机器人多种应用的环境配置,如机械臂运动规划、移动机器人导航、机器人SLAM等。

[0015] 4) 开源且免费。ROS系统的开源特性鼓励更多的人贡献自己的工作。

[0016] 本发明是基于ROS系统开发的,正是为了利用ROS系统的诸多特性,减小机械臂运动规划的实现难度,降级机械臂的应用门槛。

发明内容

[0017] 针对现有技术存在的缺陷或不足,本发明旨在提出一种基于ROS系统的单目视觉定位及机械臂抓取实现方法,可有效解决机械臂环境适应性差、开发使用难度高等不足。对于机械臂的视觉接入、目标检测、图像处理、机械臂运动规划等本发明给出了一整套的解决方案。

[0018] 实现本发明目的的技术解决方案为:

一种基于ROS系统的单目视觉定位及机械臂抓取实现方法,包括以下步骤:

步骤1:安装相机,并将所述相机连接至上位机;相机可采用USB相机,一般安装于抓取物体的上方或侧上方,要求相机的具体安装位置可确定以及具有良好的机器视角(有利于实现物体的探测),并由此获得相机所在的坐标系。

[0019] 步骤2:配置所述相机在ROS系统下的使用环境;

步骤3:所述上位机在ROS系统下进行图像处理,包括:

步骤31:读取所述相机的矫正数据,并选择要采用的识别算法;

步骤32:定位标记识别,给出空间位置信息;

步骤33:将获得的空間位置信息传递给MoveIt模块进行运动规划。

[0020] 在ROS系统下进行图像处理的程序在结构上分为:图像检测部分、定位标记识别算法部分和空间位置数据处理部分。所谓图像检测就是编写ROS节点来订阅话题usb_cam/image_raw,USB_CAM将会将图像信息发布到这个话题上。

[0021] 图像检测部分用于订阅相机驱动发布的图像信息及相机矫正参数等;

定位标记识别算法分为两种:基于ARToolKit的特定标记识别算法和基于OpenCV的标记识别算法;

空间位置数据处理部分的作用是将识别算法得到的空间位置信息进行统一化处理,将位置信息同一转换成四元数的表达形式,通过MoveIt提供的C++编程接口传递给MoveIt 初始化程序模块。

[0022] 步骤4:所述上位机在ROS系统下进行机械臂运动规划;

步骤5:将获得的机械臂运动信息队列传递至下位机;

步骤6:所述下位机接收机械臂运动信息队列并解析,同时驱动机械臂按照规划的路径执行运动和抓取,并将机械臂的实时位姿回传至所述上位机。

[0023] 进一步的,所述步骤2中配置相机的使用环境包括:

步骤21:在ROS系统下配置USB_CAM对所述相机进行驱动,同时将所述相机采集的图像信息发布出去;一般通过话题usb_cam/image_raw将相机采集的图像信息发布出去。

[0024] 步骤22:在ROS系统下对所述相机进行标定;一般采用camera_calibration对相机进行标定。

[0025] 步骤23:执行视觉标识训练。

[0026] 更进一步,所述执行视觉标识训练的过程为:

对于基于ARToolKit的特定标记识别算法,利用在线工具“Tarotaro”训练或者使用ARToolKit提供的mk_patt的高线工具进行训练;

对于基于OpenCV_ArUco的识别算法,则利用drawMarker()函数进行创建和训练标识图案。

[0027] 进一步的,所述步骤32选择的识别算法为基于ARToolKit的识别算法或基于OpenCV_ArUco的识别算法:

如果选择基于ARToolKit的识别算法,程序处理过程为:读取预先导入的标识图信息,所述相机获取实时的图像,寻找可靠度最高的标识并定位相机观察到的标识,在标识上放置系统原点,添加ARToolKit 帧频计数器,利用不同帧下标识图的位置变化预估标识图空间中的位置并实时发布,同时利用OpenGL在标识图标上绘制坐标系;

如果选用基于OpenCV_ArUco的识别算法,首先在ROS系统下配置cv_bridge将所述相机获得的sensor_msgs/Image类型图像数据转换成OpenCV库可识别的cv::Mat类型图像数据。

[0028] 然后,获取图像中所有的可疑标识,并依次提取标识图像中的标识位;

其次,利用透视变换获取标准标识图,然后利用OTSU二值化算法读取标识位的值,并将所述标识位的值与预先训练的标识图像作比较来实现标识图识别;

再次,对已识别的标识图进行空间位姿估计,并给出已识别的标识图的空间位置信息。

[0029] 最后,根据预先设定的相机坐标系和机械臂坐标系转换矩阵,获取带有标识图的物体相对于机械臂的空间位置信息,即所述空间位置信息。

[0030] 两种识别算法均以ROS节点的形式进行图像处理,给出的均为定位标记的空间位置信息。

[0031] 进一步的,所述步骤4上位机在ROS系统下进行机械臂运动规划的过程为:

步骤41:采用urdf编写机械臂模型描述文件,用于机械臂的建模;根据机械臂的实际情况对机械臂进行模型创建,在ROS系统中机械臂的模型使用基于XML脚本语言进行描述和创建。

[0032] 步骤42:利用ROS系统下的MoveIt 初始化工具包(MoveIt Setup Assistant Tool)调用所创建的机械臂描述模型,设置其运动学求解器为KDL Kinematics Plugin,默认运动算法规划库为OMPL。

[0033] 步骤43:机械臂建模;设计机械臂运动规划关节组,关节组分为手臂组(arm group)和末端执行组(grabber group)。

[0034] 步骤44:读取图像处理获得的所述空间位置信息,利用MoveIt模块进行路径规划,然后将规划的机械臂运动信息队列发布出去。

[0035] 进一步的,所述上位机和下位机之间的通信,采用Socket通信方式.Socket通信实质是在ROS中编写一个节点程序将MoveIt运动规划出的运动信息通过无线通信的方式发送

给机械臂所在的下位机,执行物品抓取动作。这里主要是指在上位机中编写Socket的Client节点。该节点订阅MoveIt 初始化程序模块产生的运动规划信息并打包发送出去。下位机中需要嵌入Socket通信的服务器程序(Service),用于接收上位机发送的运动规划信息,并将运动信息解析成机械臂可执行的运动消息队列,同时下位机驱动机械臂执行相对应的动作实现物品的抓取和放置等。

[0036] 进一步的,所述通信过程为:

步骤71:消息服务器模块初始化,用于读取MoveIt模块发布的运动规划信息队列;

步骤72:初始化Socket通信信息,并将信息队列放置在发送缓冲区,在上下位机进行通信时发送给下位机;

步骤73:下位机接收到数据后进行运动信息解析,驱动机械臂按照规划动作执行抓取;

步骤74:在执行规划动作的同时,机械臂上的位置传感器将机械臂的实际位姿通过Socket发送给上位机。

[0037] 本发明的一种基于ROS系统的视觉定位及机械臂抓取实现方法与现有技术相比,其显著优点在于:首先本发明的技术方案通过引入机器视觉作为机械臂的核心探测器件可以大大提高机械臂的适应能力,对于不同的物品,只要在机械臂的工作空间之内且物品在机械臂的负载范围之内,均可以执行抓取和放置动作;其次,本方案采用分布式的系统框架,将上位机和下位机分开,可以有效利用上位机超高的计算能力和图像处理能力,同时有利于保证下位机的实时性;再次,本发明提出的方案是基于ROS操作系统的,充分利用ROS系统的丰富软件包,实现机械臂运动规划的快速配置,大大降低了机械臂控制的门槛;最后,本发明提出的整体解决方案可以轻松的实现机械臂的布局,方便拓展为单上位机多机械臂协同工作,降低机械臂的使用成本,应用前景广泛。

附图说明

[0038] 图1为本发明提出的一种基于ROS系统的单目视觉定位及机械臂抓取实现方法的实现环境示意图;

图2为本发明提出的一种基于ROS系统的单目视觉定位及机械臂抓取实现方法的系统工作流程图;

图3为本发明提出的上位机在ROS系统下进行图像处理的工作流程图;

图4为本发明提出的上位机在ROS系统下进行机械臂运动规划的工作流程图;

图5为本发明提出的上位机与下位机进行通信的工作流程图。

具体实施方式

[0039] 如图1所示,根据本发明的较佳实施例,基于ROS系统的单目视觉定位及机械臂抓取实现方法,其实现环境包括上位机、下位机和通信环境。上位机作为图像检测和图像处理的主体,下位机作为接受命令和执行抓取的主体,共同协作完成机械臂抓取任务。

[0040] 参考图1所示的实现环境示意图,本实施例的实现环境有以下几个部分组成:

1)USB相机。相机放置在待抓取物体的上方或斜上方,以拍摄视角清晰无遮挡为最佳,且需要明确摄像机所在的坐标系(图1中所示坐标系1)。

[0041] 2)上位机。上位机需要安装有ROS操作系统(基于Linux),它是整个系统的“大脑”,

它的主要作用有：驱动USB相机完成图像采集和传输、图像处理、运动规划以及运动信息发送。

[0042] 3) 下位机。下位机是指机械臂的驱动控制部分，他的主要作用是接收运动控制信息、驱动机械臂、机械臂位姿传感检测，位姿发送。

[0043] 4) 机械臂。机械臂是本系统的执行部分，这里要求机械臂具有五个以上的自由度且机械臂具有末端执行器(吸盘或者末端夹持器)，不同的机械臂秩序在机械臂建模的时候根据实际情况调整即可。除此之外，机械臂的位置必须确定且建模时以机械臂的所在位置(图1中所示坐标系2)为基础。

[0044] 5) 局域无线网。这里Socket通信要求在实现环境中有无线网络，上下位机在相同的域段实现通信。

[0045] 6) 待抓取物体及定位标记。对待抓取物体的要求为：放置在机械臂的运动空间之内且小于机械臂的额定负载。定位标是指记用于视觉识别和定位的具有特定形状要求的标识图，对于不同的算法，标识图像则有所不同。

[0046] 如图2所示，本发明提出的一种基于ROS系统的单目视觉定位及机械臂抓取实现方法的系统工作流程图。

[0047] 整个实现流程分为上位机配置和下位机配置两个部分，下面进一步说明：

21) 在上位机的ROS系统内进行相机驱动，本实例采用的驱动节点程序为usb_cam，这一节点将驱动相机并将相机采集的图像发布在usb_cam/image_raw话题上。

[0048] 22) 利用ROS系统的camera_calibration对该相机进行标定并保存矫正数据。

[0049] 23) 执行视觉标识训练。本发明提出的两种识别算法的标识训练方法如下：

对于基于ARToolKit的特定标记识别算法，可以利用在线工具“Tarotaro”训练或者使用ARToolKit提供的mk_patt的离线工具进行训练；

对于基于OpenCV_ArUco的识别算法，则利用drawMarker()函数进行创建和训练标识图案。

[0050] 3) 将相机获得图像信息进行处理。

4) 将获得的空間位置信息传递给MoveIt 初始化程序模块进行运动规划。

[0051] 5) 将运动规划获得的机械臂运动信息队列通过Socket通信传递至下位机。

[0052] 6) 下位机接收机械臂运动信息队列并解析，同时驱动机械臂按照规划的路径执行运动和抓取，并将机械臂的实时位姿回传至上位机。

[0053] 如图3所示，上位机在ROS系统下进行图像处理的工作流程图，整个流程是在一个ROS节点上完成的，进一步说明如下：

首先，读取相机的矫正数据，并选择要采用的识别算法。

[0054] 如果选择基于ARToolKit的特定标记识别算法，程序处理过程为：读取预先导入的标识图信息，USB相机获取实时的图像，寻找可靠度最高的标识并定位相机观察到的标识，在标识上放置系统原点，添加ARToolKit 帧频计数器，利用不同帧下标识图的位置变化预估标识图空间中的位置并实时发布，同时利用OpenGL可以在标识图标上绘制坐标系。

[0055] 如果选用基于OpenCV_ArUco的识别算法，首先需要在ROS系统下配置cv_bridge将ROS系统下相机获得的sensor_msgs/Image类型图像数据转换成OpenCV库可识别的cv::Mat类型图像数据。

[0056] 然后,获取图像中所有的可疑标识,并依次提取标识图像中的标识位。

[0057] 利用透视变换获取标准标识图,然后利用OTSU二值化算法读取出标识位的值并将其与预先训练的标识图像作比较来实现标识图识别。

[0058] 对识别的标识图进行空间位姿估计,并给出空间位置信息。

[0059] 然后,根据预先设定的相机坐标系和机械臂坐标系转换矩阵,获取带有标识图的物体相对于机械臂的空间位置信息。

[0060] 最后,编写MoveIt接口程序,将空间位姿信息格式化成为四元数形式并利用MoveIt模块提供的API传递给MoveIt 初始化程序模块进行运动规划。

[0061] 图4所示为本发明提出的上位机在ROS系统下进行机械臂运动规划的工作流程图。进一步说明如下:

首先,用urdf编写机械臂模型描述文件,用于机械臂的建模。

[0062] 然后,用ROS系统下的MoveIt 初始化工具包(MoveIt Setup Assistant Tool)调用所创建的机械臂描述模型。

[0063] 进一步,创建步骤依次为:碰撞检测设置,虚关节设置,机械臂规划关节组设置,(其运动学求解器为KDL Kinematics Plugin),机械臂初始位置设置,机械臂末端执行器设置,被动关节设置,最后生成MoveIt 初始化程序模块(默认运动算法规划库为OMPL)。

[0064] 最后,读取图像处理获得的空间位置信息,利用MoveIt 初始化程序模块进行路径规划然后将规划的机械臂运动信息队列发布出去。

[0065] 图5所示为本发明提出的上位机与下位机进行通信的工作流程图,本流程是在一个ROS节点上完成的,进一步说明如下:

首先初始化消息服务器程序初始化,用于读取MoveIt 初始化程序模块发布的运动规划信息队列;

然后初始化Socket通信(TCP信息),并将数据队列放置在发送缓冲区,在上下位机进行通信时发送给下位机;

下位机接收到数据后进行运动信息解析,驱动机械臂按照规划动作执行抓取;

在执行规划动作的同时,机械臂上的位置传感器(如角度传感器、编码器) 等将机械臂的实际位姿通过Socket发送给上位机。

[0066] 综上所述,本发明与现有技术相比,其显著优点在于:本发明的技术方案采用分布式设计,既有利于利用上位机的处理能力,又方便拓扑为多机械臂协作;本发明提出的基于视觉的物体定位方法适应于抓取不同的物体,对物体初始位置要求较低;本发明提出的机械臂运动规划方法充分利用ROS系统的特点,配置简单,方便实用;本发明提出的整体解决方案采用无线通信,布局灵活,可适用于不同的应用场景。

[0067] 虽然本发明已以较佳实施例揭露如上,然其并非用以限定本发明。本发明所属技术领域中具有通常知识者,在不脱离本发明的精神和范围内,当可作各种的更动与润饰。因此,本发明的保护范围当视权利要求书所界定者为准。

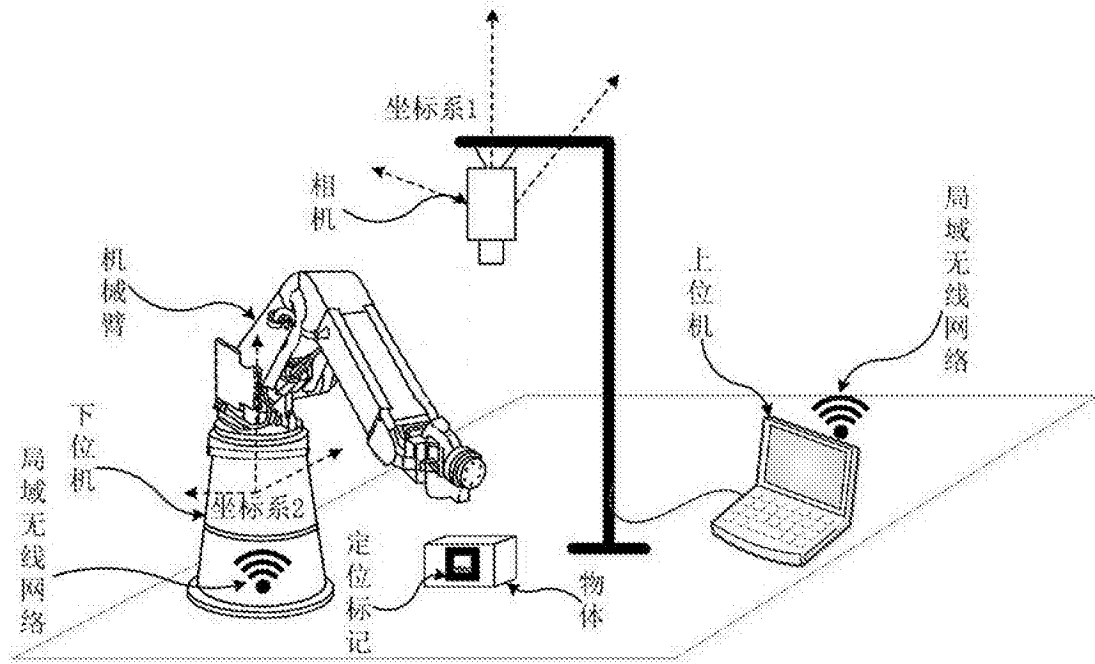


图1

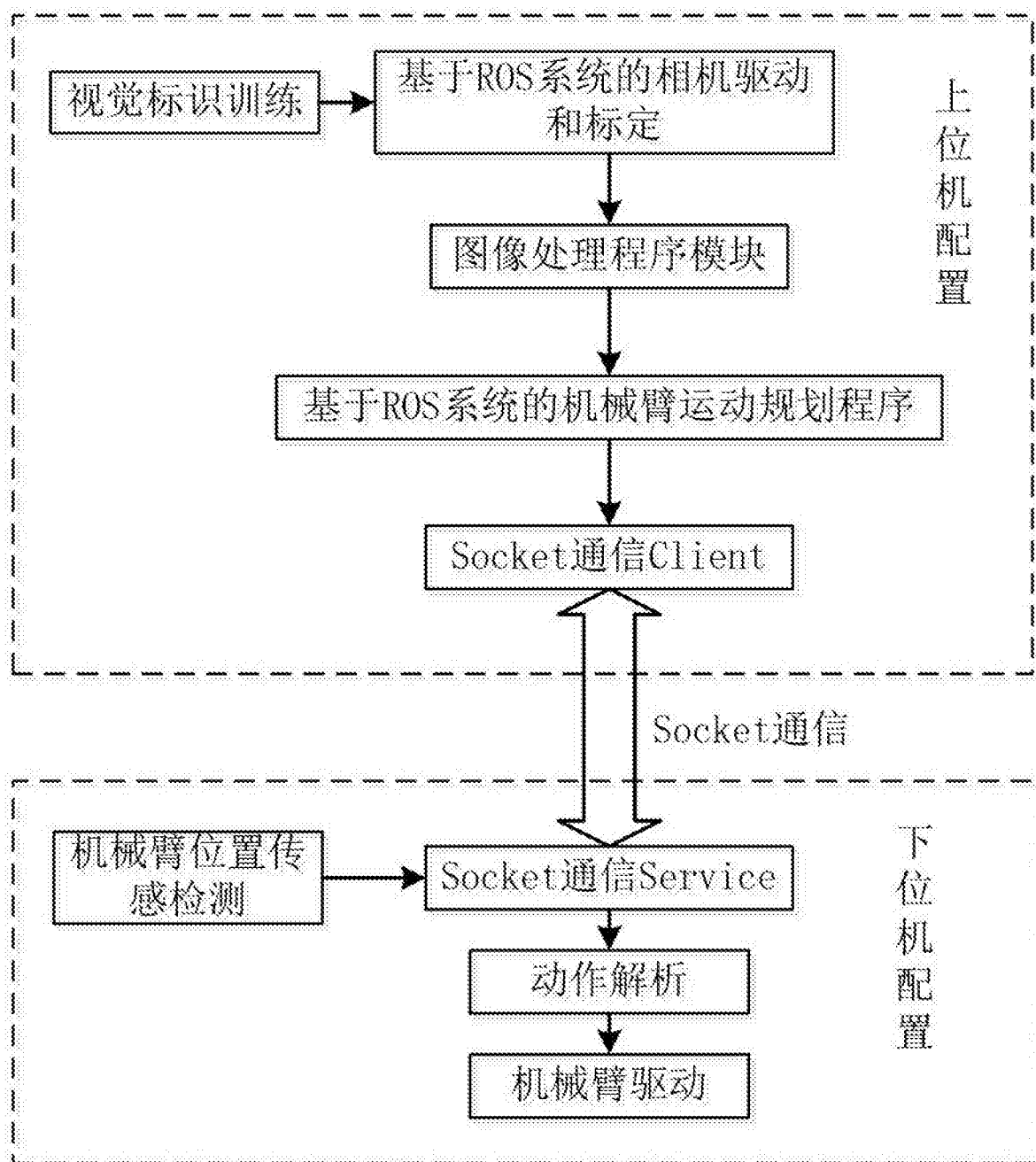


图2

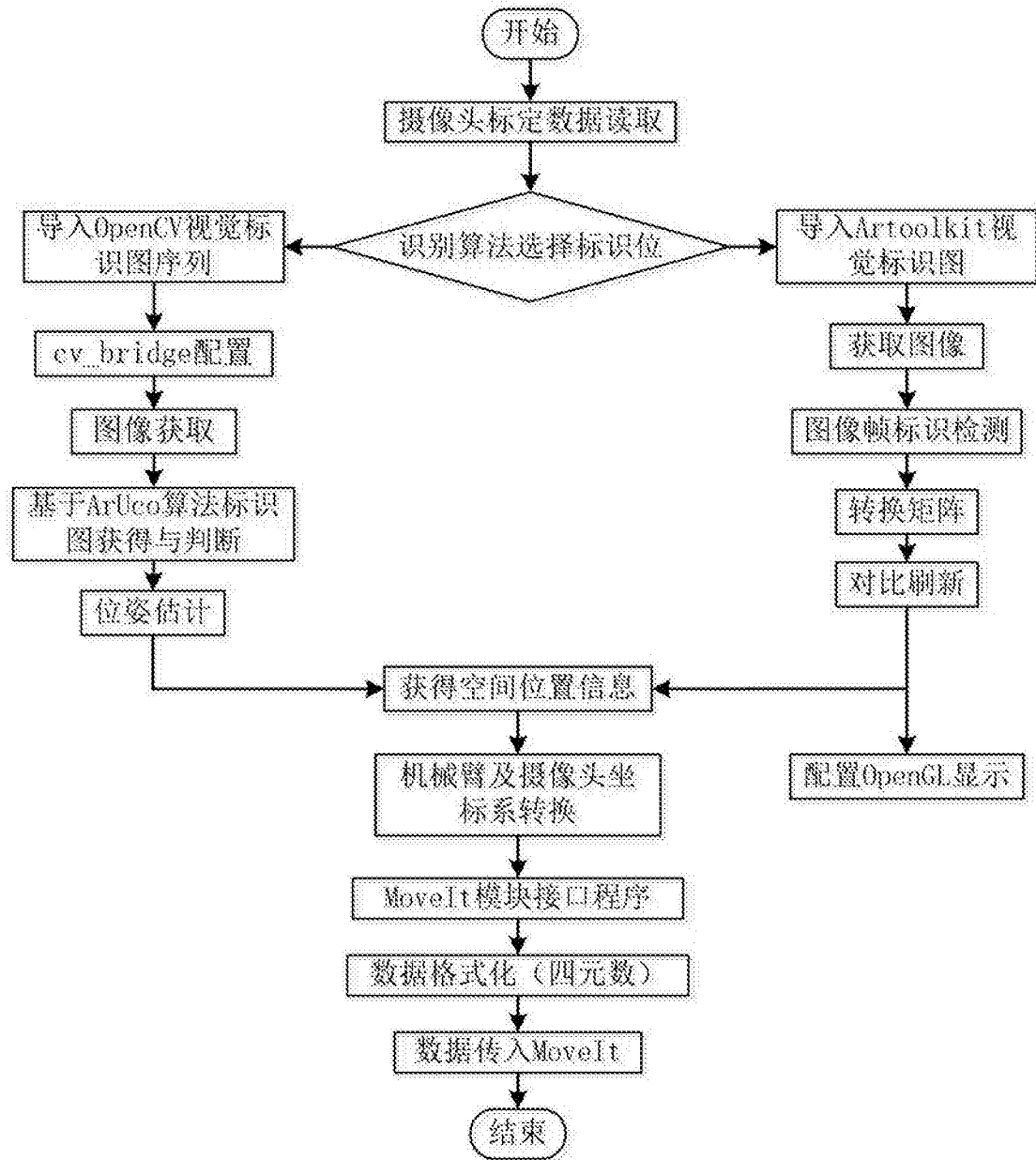


图3

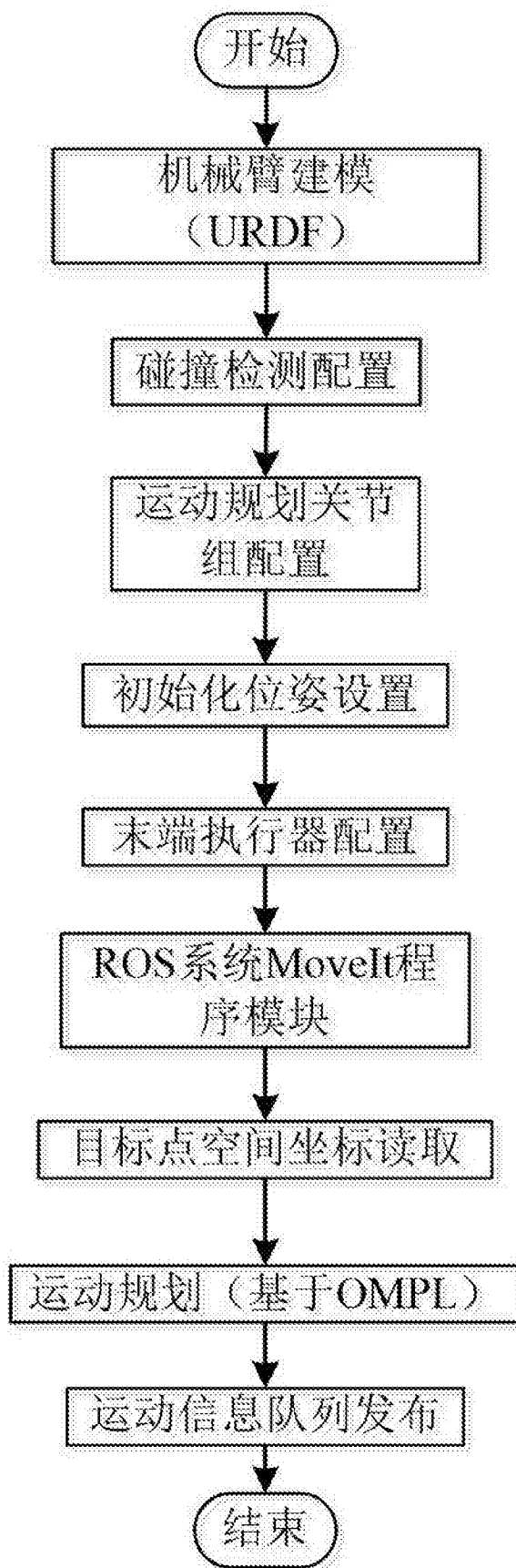


图4

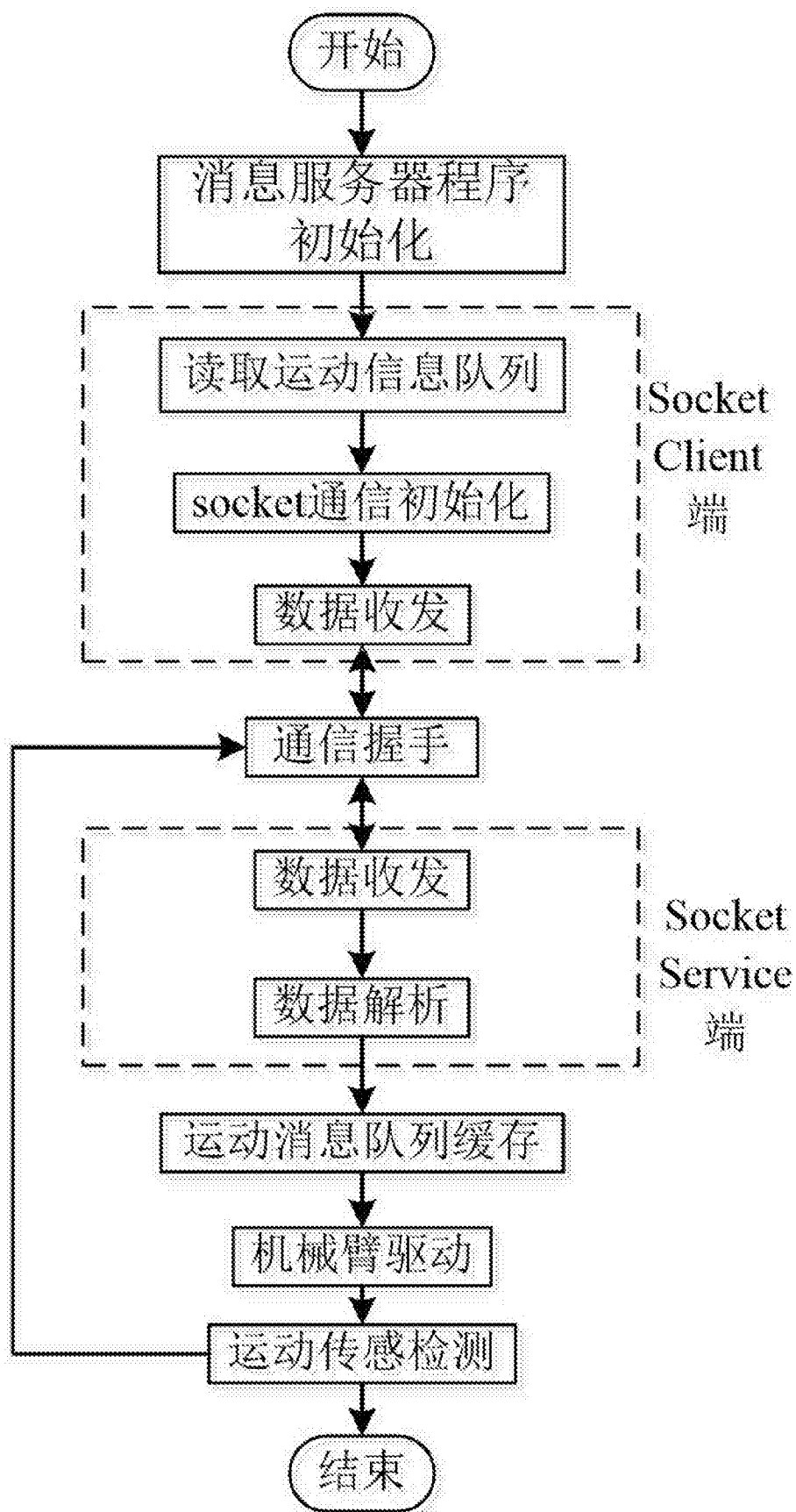


图5