

# 基于虚拟现实技术的移动机器人视觉伺服控制系统设计

田锦秀

(西安城市建设职业学院 机电信息学院, 西安 710114)

**摘要:** 为了有效确保移动机器人视觉伺服控制效果, 提高移动机器人视觉伺服控制精度, 设计了基于虚拟现实技术的移动机器人视觉伺服控制系统; 通过三维视觉传感器和立体显示器等虚拟环境的 I/O 设备、位姿传感器、视觉图像处理以及伺服控制器元件, 完成系统硬件设计; 从运动学和动力学两个方面, 搭建移动机器人数学模型, 利用标定的视觉相机, 生成移动机器人实时视觉图像, 通过图像滤波、畸变校正等步骤, 完成图像的预处理; 利用视觉图像, 构建移动机器人虚拟移动环境; 在虚拟现实技术下, 通过目标定位、路线生成、碰撞检测、路线调整等步骤, 规划移动机器人行动路线, 通过控制量的计算, 实现视觉伺服控制功能; 系统测试结果表明, 所设计控制系统的位置控制误差较小, 姿态角和移动速度控制误差仅为  $0.05^\circ$  和  $0.12\text{ m/s}$ , 移动机器人碰撞次数较少, 具有较好的移动机器人视觉伺服控制效果, 能够有效提高移动机器人视觉伺服控制精度。

**关键词:** 虚拟现实技术; 移动机器人; 视觉检测; 伺服控制系统

## Design of Visual Servo Control System for Mobile Robot Based on Virtual Reality Technology

TIAN Jinxiu

(Institute of Mechanical and Electrical Information, Xi'an Urban Architectural College, Xi'an 710114, China)

**Abstract:** To effectively ensure the visual servo control performance of mobile robots and improve the precision of mobile robot visual servo control, a virtual reality-based mobile robot visual servo control system is designed. The system hardware is composed of 3D vision sensors, stereo displays, I/O devices in virtual environments such as stereoscopic display, pose sensors, vision image processors, and servo controller components. From the aspects of kinematics and dynamics, a mathematical model of the mobile robot is established. The calibrated vision cameras are used to generate the real-time visual images of the mobile robot, using image filtering and distortion correction techniques to complete image preprocessing. The virtual mobile environment for mobile robot is constructed based on the visual images. Under virtual reality technology, the mobile robot's action trajectory is planned through the steps of target localization, path generation, collision detection, and path adjustment. By calculating the control variables, the visual servo control function is achieved. The test results show that the position control error of the designed control system is small, with angular and velocity control errors of only  $0.05^\circ$  and  $0.12\text{ m/s}$  respectively. The mobile robot has fewer collisions and exhibits a good visual servo control performance, effectively improving the precision of mobile robot visual servo control system.

**Keywords:** virtual reality technology; mobile robot; visual inspection; servo control system

## 0 引言

移动机器人是指能够在空间中自主移动的机械设备, 主要通过机械或电子控制系统实现移动和导航, 可以根据任务需求装配不同的功能模块<sup>[1]</sup>。移动机器人广泛应用于工业、医疗、家庭服务、安保等领域, 可以自主执行任务, 提高效率和安全性。随着人工智能和机器人技术的不断发展, 移动机器人的应用范围和任务也越来越广泛, 但是在复杂和不确定的环境中, 移动机器人的运动控制和路径规划仍是一个难以解决的问题。视觉伺服控制可以通过机器视觉技术实现对移动机器人姿态和位置的实时检测和估计, 从而实现移动机器人的精确控制和路径规划。除此之外,

视觉伺服控制还可以帮助机器人克服环境中的障碍物, 在特定的环境中进行定位和导航, 提高移动机器人的工作效率和精确度。为此, 研究移动机器人视觉伺服控制具有重要意义。

目前相关领域学者针对机器人视觉伺服控制系统进行研究, 文献 [2] 设计了工业机器人手眼视觉伺服控制系统。采用 Kalman 滤波器对复合图像雅克比矩阵进行在线估计, 利用 Lyapunov 原理, 构建视觉伺服控制器, 从而有效跟踪图像运动目标, 并通过平面二连杆机器人仿真实验分析验证了该系统的可行性。文献 [3] 设计了基于雅克比矩阵的高压带电作业机器人视觉伺服控制系统。采用雅可比

收稿日期: 2023-05-17; 修回日期: 2023-06-28。

作者简介: 田锦秀(1977-), 女, 大学本科, 讲师。

引用格式: 田锦秀. 基于虚拟现实技术的移动机器人视觉伺服控制系统设计[J]. 计算机测量与控制, 2023, 31(9): 150-156, 189.

矩阵，计算作业机器人执行器操作空间和关节速度间的关系，以及执行速度和视觉图像特征变化间的关系，将其结合得到复合雅克比矩阵，再利用 PID 控制器，构建机器人视觉伺服控制系统，实现参数自适应调整，该系统具有较好的控制性能。但上述两种系统仍存在控制效果较差、控制精度低的问题。

虚拟现实技术是一种通过计算机生成的模拟环境，使用户能够与虚拟环境进行交互和沉浸式体验的技术，其核心目标是提供一种逼真的、交互性强的虚拟体验，虚拟现实技术在许多领域都有广泛的应用，如游戏和娱乐、教育和培训、医疗保健、建筑和设计、航空航天、军事仿真等。为此，设计一种基于虚拟现实技术的移动机器人视觉伺服控制系统，以期能够提升系统的视觉伺服控制精度。

## 1 移动机器人视觉伺服控制硬件系统设计

### 1.1 虚拟环境的 I/O 设备

虚拟现实技术的运行需要以真实环境作为参考样本，并利用输入设备将真实环境中的图像信号和姿态信号输入到虚拟环境中，并通过输出设备输出虚拟成像结果。虚拟现实技术选择三维视觉传感器作为虚拟环境的输入设备，该设备由成像芯片和镜头等部分组成，其中成像芯片选用的是 CMOS 芯片，设置三维视觉传感器的运行速度与图像刷新的速度一致。利用三维跟踪器，可以使虚拟现实用户的运动与视觉系统进行再匹配与追踪。将虚拟环境的输入设备与移动机器人相连接，虚拟环境输入设备的连接电路如图 1 所示。

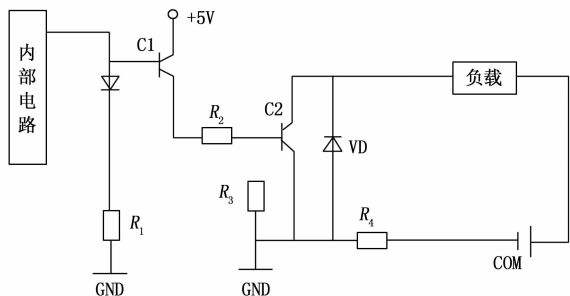


图 1 虚拟环境输入设备的连接电路图

图 1 显示了虚拟现实技术中所使用的三维视觉传感器的连接方式，系统中设置的虚拟环境输入设备共两个，用来模拟移动机器人的双目视觉。虚拟现实环境的输出设备选择立体显示器和音响设备，用两个显示器分别显示图像，这两块屏幕上的影像是由计算机控制，屏幕上的影像有细微的差异。移动机器人合成两幅影像，从而得到更好地沉浸式视觉效果。

### 1.2 位姿传感器

位姿传感器用来对控制移动机器人实时运动状态进行检测，为控制量的计算提供初始数据。为保证传感器能够实现多维姿态数据的采集，采用多传感器组合的方式形成位姿传感器，其结构如图 2 所示。

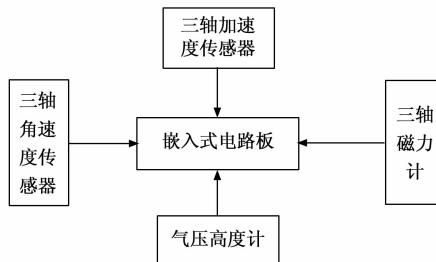


图 2 位姿传感器结构图

设计系统使用的传感器包括三轴加速度传感器、三轴角速度传感器和三轴磁力计三个部分，将传感器采集的原始数据传送至嵌入式板上，然后对其进行姿态分析，从而获得当前模块的俯仰角、横滚角、航向角、气压、高度和温度等数据，系统位姿传感器通常会保留一个通信接口，它可以将分析到的数据信息发送到 PC、单片机等设备中<sup>[4]</sup>。为了满足系统的功能需求，在机器人移动时，必须要对其进行实时的偏航角信息进行检测，以 4~7 V 为电源，工作电流大约 40 mA，同时还保留了 2 个 UART 接口，数据传输率为 115 200 bps，可以用 USB 转 TTL 模块将数据发送至 PC 机上<sup>[5]</sup>。通过选取合适的传感器，可以获得被测机器人的位姿数据，然后再由位姿资料处理装置 STM32F103T8 进行位姿数据的处理，从而获得可用的位姿数据。

### 1.3 视觉图像处理器

在移动机器人视觉伺服控制过程中，需要采集大量的视觉图像，为视觉图像处理工作提供硬件支持。首先需要确定采集的图像类型、分辨率、帧率以及采集频率等参数，然后采用 Sony IMX219 图像传感器采集图像信号，最后选择一台工业控制计算机，对图像序列进行分析和处理，其使用 Intel i3 处理器，最高主频 1.6 GHz，8 GB 的内存，确保了图像处理的速度，并且保留 2 个网口，2 个 USB3.0 接口，4 个 COM 接口，4 个 COM 接口，这使工业控制计算机能够与大量的外设进行连接。

### 1.4 伺服控制器

伺服控制器是移动机器人视觉伺服控制系统控制功能的执行元件，控制器由驱动设备、处理器等部分组成。由于虚拟现实环境的构建，使得移动机器人的运动不受空间环境的限制，扩大了移动机器人的活动范围<sup>[6]</sup>。选用直流电机作为移动机器人的动力执行机构，机器人对运动速度的控制处于开环状态，需要根据姿态传感器反馈的机器人实时运动速度，完成机器人运动速度的闭环控制。在载荷作用下，能够维持恒定的速度，从而实现了对机械臂相对高精度的运动控制。伺服驱动电机具备完备的电流保护，可有效防止因过流或过温而导致的硬体损坏。伺服控制器内部的处理器选择 AT91SAM7S256 芯片，该芯片使用 32 位 RISC 架构，可快速实现比较复杂的控制算法。芯片具有 256 kB 的片上 FLASH 和 64 kB 的 RAM 存储器空间，具备运行嵌入式实时操作系统  $\mu\text{c}/\text{OS-II}$  的能力<sup>[7]</sup>。芯片中有大

量的周边元件,如 USART、SPI、TWI 等,同时还提供了多种可编程的时钟,便于对回路进行定时操作。控制芯片采用 LQFP 封装,仅有 64 个管脚,许多管脚用作供电和信号线,可用于一般 I/O 接口较少<sup>[8]</sup>。针对这一缺陷,采用可编程逻辑器件 EPM570T144I5 来进行输入输出扩展。ARM7 通过串行的接口,能够控制 CPLD 上的一组通用 I/O 接口,并且能够将每个端口与内部的脉冲发生器连接,使其发出设定宽度和相应的脉冲,来控制伺服驱动电机的运作。伺服控制器元件连接如图 3 所示。

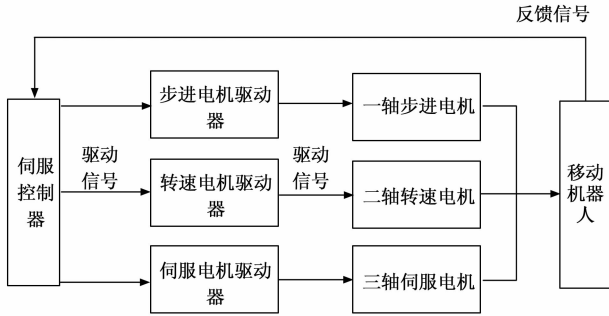


图 3 伺服控制器元件连接示意图

另外,在实际控制过程中,控制器、传感器以及移动机器人之间采用网络进行通信发送命令。

## 2 移动机器人视觉伺服控制系统软件功能设计

在硬件设备的支持下,对移动机器人的视觉伺服控制功能进行设计,从位姿、运动参数等方面实现对移动机器人的控制,保证机器人能够在虚拟现实环境中按照指定方式完成相应的移动任务。以立体标定方法、视觉图像成像原理、图像预处理方法、图像拼接与三维重构方法、虚拟现实技术的碰撞检测技术、伺服控制逻辑为主要设计方法,实现移动机器人视觉伺服控制系统软件功能设计。

### 2.1 搭建移动机器人数学模型

在设计系统控制功能之前,首先需要确定移动机器人的基本运动原理,因此,从运动学和动力学两个方面,构建机器人的数学模型。假设移动机器人存在纯滚动无滑动运动,则移动机器人的运动学模型可以表示为:

$$\begin{cases} x(t) = -v(t)\sin\theta(t) \\ y(t) = v(t)\cos\theta(t) \\ \dot{\theta}(t) = \omega(t) \\ v(t) = Q^{-1}\vartheta_{\text{line}}(t) \\ \omega(t) = M^{-1}\vartheta_{\text{horn}}(t) \end{cases} \quad (1)$$

式中,  $(x(t), y(t))$  为  $t$  时刻的移动机器人位置坐标,  $\theta(t)$  表示移动机器人的姿态角,  $v(t)$  为移动机器人在  $t$  时刻的线速度<sup>[9]</sup>,  $\omega(t)$  为移动机器人在  $t$  时刻的角速度,  $Q$  为机器人载荷总重量,  $M$  为机器人惯性量度,  $\vartheta_{\text{line}}(t)$  和  $\vartheta_{\text{horn}}(t)$  分别为机器人的线转矩和角转矩<sup>[10]</sup>。在移动状态下,移动机器人的动力学特性满足如下规律:

$$\begin{cases} \theta(t)(F_{\text{Drive-Front}} + F_{\text{Drive-Rear}} - 2F_f) = m\dot{v}(t) \\ \frac{x(t)y(t)(F_{\text{Drive-Front}} - F_{\text{Drive-Rear}})d}{2\omega(t)} = M \end{cases} \quad (2)$$

式中,  $F_{\text{Drive-Front}}$  和  $F_{\text{Drive-Rear}}$  分别为驱动电机对前轮和后轮的驱动力,  $F_f$  为移动机器人与地面之间的摩擦力,  $m$  为移动机器人的质量,  $d$  为前轮与后轮之间的距离<sup>[11]</sup>。通过运动学和动力学模型的融合,得出移动机器人数学模型的搭建结果。

### 2.2 标定移动机器人视觉相机

为保证移动机器人采集视觉图像的精准度,在图像采集之前,采用立体标定方式,标定移动机器人的视觉相机。针对双目移动机器人,两个视觉相机生成图像需要经过数学变换的方式转换才能分布在同一平面上,其中,数学变换过程需要使用旋转矩阵和平移矩阵。对于空间中任意一点  $P$ ,其在两个视觉相机中的成像结果可以分别表示为  $P'_1$  和  $P'_2$ ,两者之间的坐标关系可以表示为:

$$\begin{cases} P'_1 = MJ_{\text{left}}(P - U_{\text{left}}) \\ P'_2 = m\dot{v}(t)J_{\text{right}}(P - U_{\text{right}}) \end{cases} \quad (3)$$

式中,  $J_{\text{left}}$  表示双目视觉相机的左侧旋转矩阵,  $U_{\text{left}}$  表示左侧平移矩阵,  $J_{\text{right}}$  表示右侧旋转矩阵,  $U_{\text{right}}$  表示右侧平移矩阵。进行立体标定时,利用视觉相机提取包含标定板的图像并进行识别,可以得到空间中点与其对应的摄像机成像平面上点的对应关系,从而得出矩阵  $Z$  和  $U$  的求解结果<sup>[12]</sup>。根据左、右视觉相机之间的实际位置距离,确定矩阵  $Z$  和  $U$  的实际值,并与求解出的矩阵结果进行比对,判断当前视觉相机的成像结果是否存在偏差,通过偏差的校正实现对移动机器人视觉相机的标定。

### 2.3 生成移动机器人实时视觉图像

利用标定完成的视觉传感器设备根据移动机器人的移动情况,生成实时视觉图像。视觉图像的成像原理如图 4 所示。

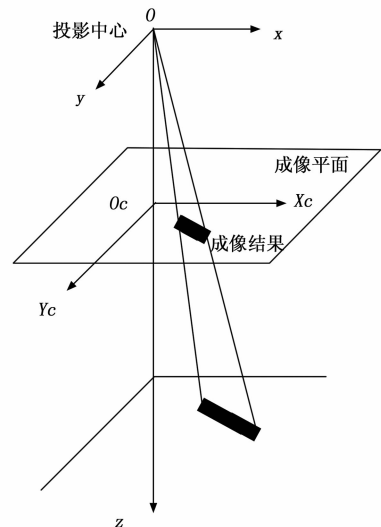


图 4 移动机器人实时视觉图像成像原理图

利用图 4 表示成像原理，假设实际空间中任意一点  $(x_{true}, y_{true}, z_{true})$  在成像空间中的成像结果可以表示为：

$$\begin{cases} x = P'_1 \frac{x_{true} \cdot z_{true}}{f} \\ y = P'_2 \frac{y_{true} \cdot z_{true}}{f} \end{cases} \quad (4)$$

式中， $f$  为视觉传感器的成像焦距。在考虑视觉相机标定结果的情况下，可以将视觉相机的成像结果表示为：

$$I(x, y) = \begin{pmatrix} \frac{f}{L_x} & -\frac{f \cot \varphi}{L_x} & x_c & 0 \\ 0 & \frac{f}{L_y \sin \varphi} & y_c & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} A_w \begin{pmatrix} x_{true} \\ y_{true} \\ z_{true} \\ 1 \end{pmatrix} \quad (5)$$

式中， $L_x$  和  $L_y$  分别为任意点在成像平面上物理长度在水平和竖直方向上的分量， $\varphi$  为成像角度， $(x_c, y_c)$  表示像素坐标系中相机感光板的中心坐标， $A_w$  表示设定相机外参矩阵<sup>[13]</sup>。移动机器人的成像视场为：

$$W = 2 \tan^{-1} \left( \frac{R}{2d_{thing}} \right) I(x, y) \quad (6)$$

式中， $R$  和  $d_{thing}$  分别为视野宽度和物距。在实际的成像过程中，设置视觉传感器的更新速度、采集频率等参数，得出移动机器人实时视觉图像的生成结果。

### 2.4 初始图像预处理

为保证移动机器人视觉伺服控制效果，以提升图像质量为目的，针对移动机器人初始采集的图像，采取一系列预处理步骤，包括灰度化、图像滤波、畸变校正等。将 RGB 图像转化成灰度图像可以有效减少数据量。对一幅彩色图像进行灰度处理，即将一幅彩色图像中的一幅图像转化为一幅灰度图像<sup>[14]</sup>。灰度图像是用来量化亮度的，一般分为 0~255 的 256 个等级，其中 0 最暗，255 最亮。初始视觉图像的灰度化处理结果为：

$$I_{Grayscale} = W(0.299 \times I_R + 0.587 \times I_G + 0.114 \times I_B) \quad (7)$$

式中， $I_R$ 、 $I_G$  和  $I_B$  分别为图像中红、绿、蓝色彩分量<sup>[15]</sup>。设计系统中采用高斯滤波处理方式，其处理过程如下：

$$I_{Gaussian}(x, y) = I_{Grayscale} \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (8)$$

式中， $\sigma$  为高斯滤波同性扩散系数。初始视觉图像中的畸变可以分为径向畸变和切向畸变两个部分，其中径向畸变是因为光线经过镜头时，在透镜的边缘附近的光会比在透镜的中央附近的光扭曲得更厉害。因此，当光线到达像面时，其初始位置发生了偏差，从而引起像面上物体的坐标发生了改变。透镜的径向失真往往导致影像出现“枕状”或“桶状”的失真。视觉图像径向畸变的校正过程为：

$$\begin{cases} x_{check} = (I_{Gaussian}(x, y) + \kappa_{check1} j^2 + \kappa_{check2} j^4 + \kappa_{check3} j^6) x \\ y_{check} = (I_{Gaussian}(x, y) + \kappa_{check1} j^2 + \kappa_{check2} j^4 + \kappa_{check3} j^6) y \end{cases} \quad (9)$$

式中， $\kappa_{check1}$ 、 $\kappa_{check2}$  和  $\kappa_{check3}$  分别为畸变校正系数， $j$  为旋转矩阵向量。另外，切向畸变的产生通常是由于摄像机中镜头和成像平面不能保持完全平行，根据切向畸变的形成原理，

确定相应的校正系数，并得出相应的切向畸变校正处理结果。将初始采集的机器人视觉图像中的所有像素代入到上述公式中，通过上述环节的处理得出移动机器人视觉图像的预处理结果。

### 2.5 构建机器人虚拟移动环境

以移动机器人实时生成的视觉图像及其处理结果为处理对象，生成相应的虚拟现实环境，虚拟现实环境构建流程如图 5 所示。

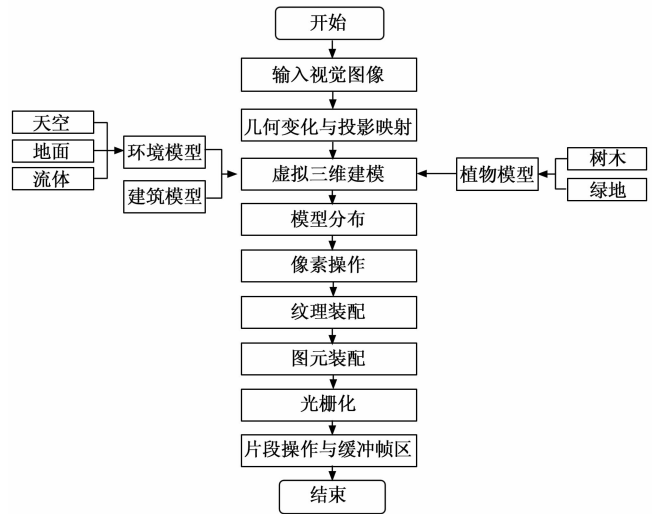


图 5 虚拟现实环境构建流程图

为保证构建的虚拟现实环境与机器人移动环境一致，采用图像拼接与三维重构的方式构建虚拟环境。在图像拼接过程中，由于移动机器人的采样频率较高，因此连续两个时刻采集的图像之间存在较大程度的重复，因此可以根据图像之间的重叠情况进行拼接处理。利用公式 (10) 提取视觉图像中任意一点的特征向量。

$$\tau(x, y) = \begin{bmatrix} \left( \frac{\partial \beta}{\partial x_{check}} \right)^2 & \left( \frac{\partial \beta}{\partial x_{check}} \right) \left( \frac{\partial \beta}{\partial y_{check}} \right) \\ \left( \frac{\partial \beta}{\partial x_{check}} \right) \left( \frac{\partial \beta}{\partial y_{check}} \right) & \left( \frac{\partial \beta}{\partial y_{check}} \right)^2 \end{bmatrix} \quad (10)$$

式中， $\beta$  为视觉图像中的像素点的亮度值， $\tau(x, y)$  为像素点的位置坐标，那么整幅视觉图像的全局特征可以表示为：

$$\tau(I) = g_{det}[\tau(x, y)] - 0.04(g_{trace}[\tau(x, y)])^2 \quad (11)$$

式中， $g_{det}$  和  $g_{trace}$  分别为局部极大值和极小值的检索函数。按照上述方式可以得出所有视觉图像的特征提取结果，并将其代入到公式 (12) 中，计算任意两幅视觉图像之间的特征匹配度。

$$s(i, k) = \frac{\tau(I_i) \cdot \tau(I_k)}{\|\tau(I_i)\| \cdot \|\tau(I_k)\|} \quad (12)$$

若公式 (12) 的计算结果高于阈值  $s_0$ ，则认为视觉图像  $i$  和  $k$  之间存在重叠情况，通过调整图像特征区域对特征匹配度进行反复计算，当计算得出  $s(i, k)$  取值为 1 时，说明当前区域为两幅视觉图像的重叠区域，将两者进行重合处理，

完成二者的拼接工作。若计算得出  $s(i, k)$  的值低于  $s_0$ ，则表示两幅图像之间不存在重叠情况，则直接进入下一组图像的匹配工作<sup>[16]</sup>。反复执行上述操作，直至移动机器人采集的视觉图像全部拼接完成为止。在二维拼接视觉图像的基础上，通过对成像点距离的分析，确定图像像素点的深度值。移动机器人对图像中任意一点实际感知深度的计算公式如下：

$$h = \frac{\xi_{\text{vision}} d_{\text{perpendicular}}^2}{d_{\text{camera}} - \xi_{\text{vision}} d_{\text{perpendicular}}} s(i, k) \quad (13)$$

式中， $d_{\text{camera}}$  和  $d_{\text{perpendicular}}$  分别表示的是移动机器人中左右两视觉传感器之间的距离以及主视传感器与目标物体之间的垂直距离，而  $\xi_{\text{vision}}$  代表图像视差。公式 (13) 的计算结果即为当前图像像素点在虚拟现实环境中的深度坐标。将视觉图像中各个物体的二维坐标与求解的深度坐标相结合，即可得出该物体在虚拟环境中的三维坐标，通过多像素的三维重构，得出机器人移动环境中所有物体结构的虚拟模型生成结果，将纹理图片贴在物体模型表面<sup>[17]</sup>。由于移动机器人处于实时运动的状态，因此相应的虚拟现实环境需要根据机器人的移动位置进行动态更新，将移动机器人车体的绝对坐标转换为参考坐标，根据虚拟现实环境中的物体与机器人之间的距离，确定对应虚拟场景是否显示，进而驱动虚拟现实场景实时更新。最终对虚拟现实场景进行渲染输出，得出机器人虚拟移动环境的搭建结果。

## 2.6 虚拟环境下规划移动机器人行动路线

### 2.6.1 目标物体的视觉定位与跟踪

通过对视觉图像特征的提取，实时检测移动机器人抓取目标的位置。根据抓取目标状态可以将其分为静态目标和动态目标，静态目标位置不会随时间变化而发生变化，定位结果如下：

$$\psi = \operatorname{argmin}_i h \sum_i \gamma_i - I_{\text{background}} \quad (14)$$

式中， $\gamma_i$  为视觉图像的边缘特征， $I_{\text{background}}$  表示的是视觉图像中的背景区域，求解结果表示目标空间位置，可以进一步展开成坐标形式。而动态目标的定位与静态目标一致，但需要定期重复定位操作，并用最新的定位结果与原始结果进行覆盖。

### 2.6.2 生成机器人初始移动路线

假设移动机器人的初始位置为  $(x_0, y_0, z_0)$ ，目标位置为  $(x_{\text{stop}}, y_{\text{stop}}, z_{\text{stop}})$ ，则初始生成的机器人移动路线可以表示为：

$$\frac{(x - x_0)}{(x_{\text{stop}} - x_0)} = \frac{(y - y_0)}{(y_{\text{stop}} - y_0)} = \frac{(z - z_0)}{(z_{\text{stop}} - z_0)} \quad (15)$$

在初始移动路线下，机器人所经过的空间路径长度为：

$$L_{\text{robot}} = \psi \sqrt{(z_{\text{stop}} - z_0)^2 + (y_{\text{stop}} - y_0)^2 + (x_{\text{stop}} - x_0)^2} \quad (16)$$

在此基础上，由机器人的初始位置至终止位置所插补步数可以表示为：

$$n_{\text{step}} = \frac{L_{\text{robot}}}{\Delta L} \quad (17)$$

式中， $\Delta L$  为等距直线插补时的步长。那么在已知虚拟环境

下移动机器人目标位置的情况下，即可生成移动机器人在虚拟环境中的初始移动路线。

### 2.6.3 利用虚拟现实技术检测碰撞状态

利用虚拟现实技术中的碰撞检测功能，判断初始生成的机器人移动路线与虚拟环境静态物体之间是否存在碰撞现象。采用层次包围盒方法执行碰撞检测任务，包围盒的每一条直线都是与其坐标轴线相平行的，它是一个最小的矩形平面<sup>[18]</sup>。基于碰撞检测的应用，常采用的基本思路是将复杂的几何体用一个简单的包围盒进行围合。设置包围盒的中心位置为：

$$C = \frac{1}{3n_{\text{plane}}} \sum_{i=1}^{n_{\text{plane}}} (b_1(i) + b_2(i) + b_3(i)) \quad (18)$$

式中， $b_1(i)$ 、 $b_2(i)$  和  $b_3(i)$  表示物体表面三角形面片三个顶点的矢量， $n_{\text{plane}}$  为包围盒包围的三角面片数。当检测两个物体的碰撞时，先判断盒周围根节点的相交情况，不能相交，那么就不会有物体与之碰撞，若相交，就会遍历根节点之下的子节点，若有相交，就会对包围盒中的两个多边形进行精确地检验。

### 2.6.4 机器人移动路线调整与更新

若检测得出初始生成的机器人移动路线与虚拟现实环境之间存在碰撞行为，则需要利用公式 (18) 调整碰撞点的位置。

$$\begin{cases} x_{\text{new}}(i) = C(x_{\text{collide}}(i) \pm \lambda_x) \\ y_{\text{new}}(i) = C(y_{\text{collide}}(i) \pm \lambda_y) \\ z_{\text{new}}(i) = C(z_{\text{collide}}(i) \pm \lambda_z) \end{cases} \quad (19)$$

式中， $(x_{\text{collide}}(i), y_{\text{collide}}(i), z_{\text{collide}}(i))$  和  $(x_{\text{new}}(i), y_{\text{new}}(i), z_{\text{new}}(i))$  表示的是调整前后碰撞路径节点的坐标值，参数  $\lambda_x$ 、 $\lambda_y$  和  $\lambda_z$  对应的是三个空间方向上的调整量<sup>[19]</sup>。按照上述方式对初始移动路线中的所有碰撞节点进行调整，并将调整后的节点接入到初始路线中，并与碰撞节点的前后节点进行连接与插补。根据机器人的实时移动位置，调整移动路线的初始位置，实现对移动路线的实时更新。

## 2.7 实现移动机器人视觉伺服控制功能

在视觉伺服控制器的支持下，按照伺服控制逻辑，生成视觉伺服控制指令，如图 6 所示。

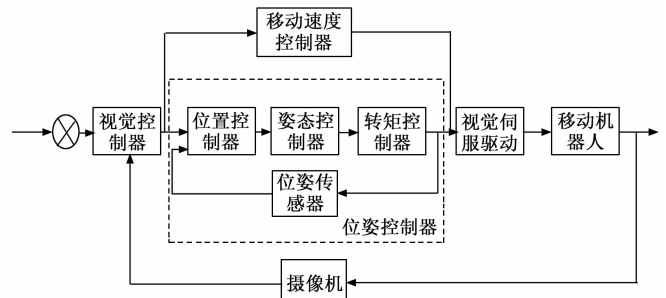


图 6 视觉伺服控制指令图

视觉伺服控制的目标就是减小移动机器人实际运动中的移动机器人参数和预期间误差，从姿态、移动速度、位置三个方面执行伺服控制任务，以移动速度控制为例，具

体控制量可以表示为:

$$\Delta v = (x_{\text{new}}(i), y_{\text{new}}(i), z_{\text{new}}(i))(v_{\text{target}} - v_t) \quad (20)$$

式中,  $v_{\text{target}}$  和  $v_t$  分别表示移动机器人的目标移动速度和  $t$  时刻的实际移动速度, 利用硬件系统中的位姿传感器可以直接得出变量  $v_t$  的具体取值。若公式 (20) 计算结果为负数, 则执行减速控制操作, 若速度控制量  $\Delta v$  取值为正, 则执行加速控制操作<sup>[20]</sup>。同理可以得出移动机器人位置与姿态的伺服控制结果。在虚拟现实环境下, 通过对视觉图像的分析, 确定控制目标参数, 并将伺服控制器输出的控制指令直接作用在移动机器人上, 直至移动机器人到达目标位置或完成目标任务。

针对移动机器人运动控制精度低的问题, 基于其运动学和动力学特征, 建立移动机器人数学模型, 采用立体标定方式标定移动机器人的视觉相机, 提高图像采集的精准度, 以此为基础生成实时视觉图像, 并对取向进行灰度化、图像滤波、畸变校正等预处理, 提高图像的质量。根据高质量图像生成相应的虚拟现实环境, 在此环境下规划移动机器人行动路线, 并及时更新移动路线, 提高机器人的伺服控制能力, 最后按照伺服控制逻辑, 生成视觉伺服控制指令, 指导移动机器人完成目标任务, 实现移动机器人视觉伺服控制系统设计。

### 3 系统测试

为了验证设计的基于虚拟现实技术的移动机器人视觉伺服控制系统中控制功能能否满足预期, 进行系统测试。此次实验分别从控制精度和控制效果两个方面进行验证, 其中控制精度测试就是测试移动机器人位置、姿态角、移动速度等运动参数与控制目标之间的误差, 而控制效果就是测试在控制系统作用下, 移动机器人与外界环境之间的碰撞情况。

#### 3.1 准备移动机器人样机

实验选择 Dr Robot 型号的移动机器人作为系统的控制对象, 该机器人带有一个可移动底盘和四个自由度铰接手臂, 能够执行多种抓取任务。Dr Robot 移动机器人自重 35 kg, 与地面之间的最大摩擦力为 30 N, 因此可以在坡度为 45° 以下的地面上自由移动。在开始试验之前, 需要对移动机器人样机的各个功能进行检测, 判断移动机器人各个元件之间是否能够协同运行、移动机器人是否能够成功接收相关控制信息、移动机器人是否能够根据控制指令执行相应任务, 在确定移动机器人样机调试成功的情况下, 执行系统测试实验的下一步操作。

#### 3.2 选择虚拟现实场景实现工具

为支持设计视觉伺服控制系统中虚拟现实技术的运行, 设置虚拟现实场景的开发工具为 Multigen creator 三维建模软件, Multigen creator 具有建模、变换、动画及物体运动模糊等功能, 其可以与 Visual C++ 等编程工具一起工作, 还能够快速地生成一个具有真实效果的大型三维场景模型, 并使用 Open Flight 作为图形数据描述格式。Open Flight 用

于告知图像生成者绘制出实时 3D 场景的时间, 是一种对场景数据进行组织的方式。将 Multigen creator 工具下载至主测计算机中, 通过解码、解压等步骤, 保证虚拟现实场景实现工具能够在主测计算机中正常运行。

#### 3.3 生成移动机器人视觉伺服控制任务

在虚拟环境中随机生成虚拟目标, 结合移动机器人样机在虚拟环境中的初始位置, 规划移动机器人的视觉伺服控制路线, 移动机器人的视觉伺服位置控制路线如图 7 所示。

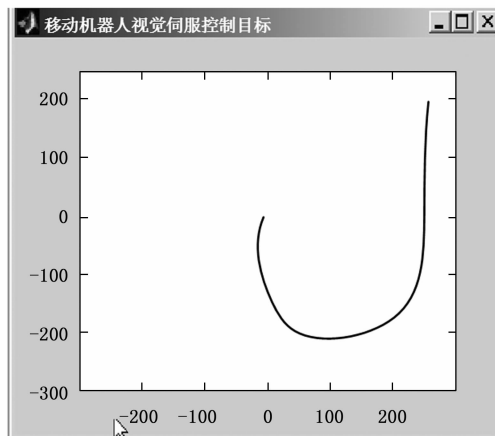


图 7 移动机器人的视觉伺服位置控制路线

在此基础上, 标记移动机器人移动过程中的实时姿态角和速度参数。按照上述方式, 生成 6 个虚拟目标, 并形成相应的 6 组视觉伺服控制任务。在控制效果测试过程中, 在相同的虚拟环境下, 生成随机障碍物, 随机障碍物可以分为静态障碍物和动态障碍物两种, 在有障碍环境下调整初始控制任务, 得出有障碍物场景下的任务生成结果。

#### 3.4 描述系统测试过程

将硬件系统中的相关元件设备安装到移动机器人内部, 将视觉传感器帧速率设置 30FPS, 分辨率设定为 640×480。通过实时图像的采集与虚拟现实技术的运行, 输出移动机器人的虚拟现实活动环境, 如图 8 所示。

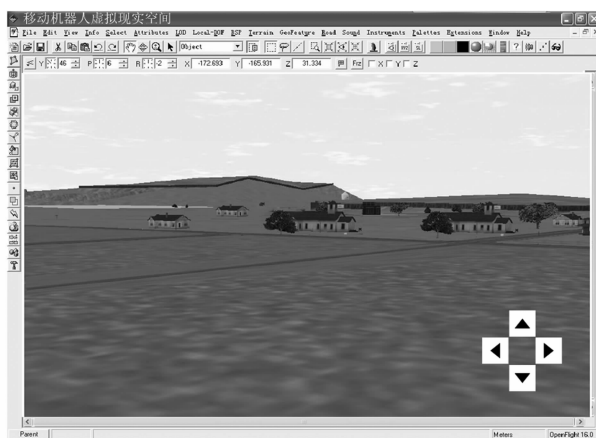


图 8 移动机器人虚拟现实活动环境生成结果

将基于虚拟现实技术的移动机器人视觉伺服控制系统的软件功能部分以程序代码的方式输入到主测计算机中, 得出控制结果。移动机器人视觉伺服控制系统运行界面如图 9 所示。

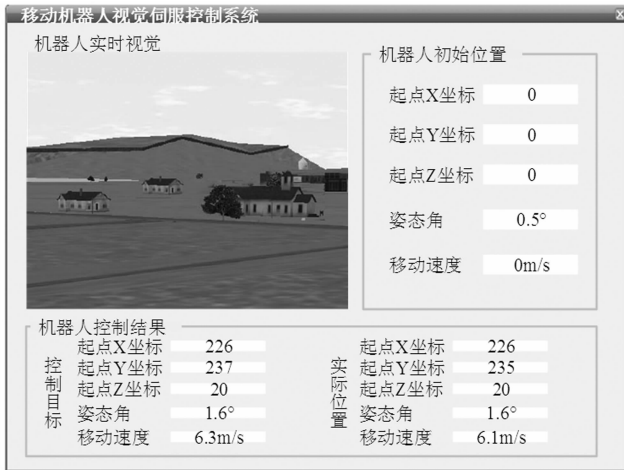


图 9 移动机器人视觉伺服控制系统运行界面

当系统检测到移动机器人与目标位置相同时, 判定移动机器人完成视觉伺服控制任务, 自动退出控制程序, 并记录机器人移动过程中的所有数据。此次系统测试实验中设置文献 [2] 系统 (基于改进 BP 神经网络的机器人视觉伺服控制系统) 和文献 [3] 系统 (基于 Udwadia-Kalaba 方法的机器人伺服控制系统) 作为对比系统, 按照上述方式完成对比系统的开发, 并得出相应的控制结果。

### 3.5 设置系统测试指标

设置位置、姿态角和移动速度控制误差作为系统控制误差的量化测试指标, 其中位置控制误差的测试结果如下:

$$\epsilon_{\text{seat}} = |x_{\text{control}} - x_{\text{target}}| + |y_{\text{control}} - y_{\text{target}}| + |z_{\text{control}} - z_{\text{target}}| \quad (21)$$

式中,  $(x_{\text{control}}, y_{\text{control}}, z_{\text{control}})$  和  $(x_{\text{target}}, y_{\text{target}}, z_{\text{target}})$  分别表示控制系统下移动机器人的实际位置和虚拟目标位置。

姿态角和移动速度控制误差的数值结果为:

$$\begin{cases} \epsilon_{\text{angle}} = |\theta_{\text{control}} - \theta_{\text{target}}| \\ \epsilon_{\text{velocity}} = |v_{\text{control}} - v_{\text{target}}| \end{cases} \quad (22)$$

式中,  $\theta_{\text{control}}, \theta_{\text{target}}, v_{\text{control}}$  和  $v_{\text{target}}$  分别为移动机器人姿态角和移动速度的控制值与目标值。另外, 控制效果的量化测试指标为碰撞次数, 其测试结果为:

$$N_{\text{collide}} = N_{\text{obstacle}} + N_{\text{Fixed objects}} \quad (23)$$

式中,  $N_{\text{obstacle}}$  和  $N_{\text{Fixed objects}}$  分别为移动机器人与障碍物以及固定物体之间的碰撞次数。最终计算得出位置、姿态角和移动速度控制误差越小, 表明系统的控制精度越高, 而碰撞次数越少, 表明系统的控制效果越好。

### 3.6 系统测试结果与分析

#### 3.6.1 控制精度

采集控制系统作用下移动机器人实时位置数据, 并将

其与位置控制目标数据输入到公式 (21) 中, 获得移动机器人位置控制误差测试对比结果如图 10 所示。

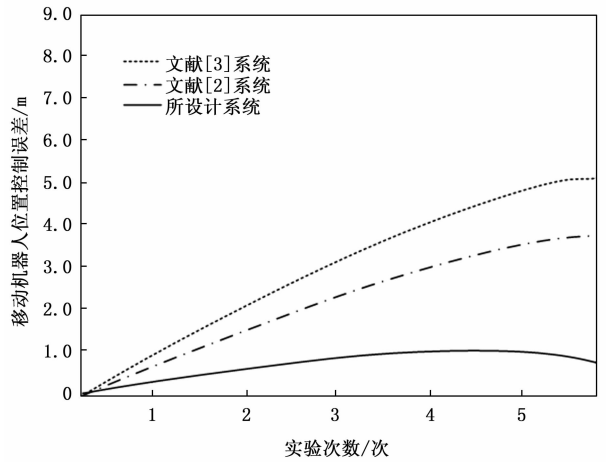


图 10 移动机器人位置控制误差测试对比结果

从图 10 中可以看出, 与文献 [2] 系统和文献 [3] 系统相比, 所设计系统的移动机器人位置控制误差较小。由此可知, 所设计系统的移动机器人视觉伺服控制精度较高。

根据上述数据, 利用公式 (22) 计算得到移动机器人姿态角和移动速度控制误差测试数据如表 1 所示。

表 1 移动机器人姿态角和移动速度控制误差测试数据表

控制任务编号	运动参数控制目标		文献[2]系统		文献[3]系统		所设计系统	
	姿态角/ (°)	移动速度/ (m/s)	姿态角/ (°)	移动速度/ (m/s)	姿态角/ (°)	移动速度/ (m/s)	姿态角/ (°)	移动速度/ (m/s)
1	1.6	6.3	1.3	6.9	1.2	6.8	1.6	6.1
2	0.4	8.2	0.8	8.8	0.6	8.5	0.4	8.1
3	0.8	10.7	0.5	10.2	0.3	10.0	0.6	10.5
4	1.6	12.1	1.1	12.5	0.9	12.4	1.6	12.0
5	2.9	9.2	2.5	9.6	2.4	9.4	2.8	9.2
6	2.8	7.7	2.1	7.3	1.8	7.1	2.8	7.6

根据表 1 中的数据可知, 文献 [2] 系统和文献 [3] 系统的平均姿态角控制误差分别为  $0.43^\circ$  和  $0.55^\circ$ , 平均移动速度控制误差均为  $0.48\text{m/s}$ , 而所设计系统的平均姿态角和移动速度控制误差分别为  $0.05^\circ$  和  $0.12\text{m/s}$ 。由此可知, 所设计系统的移动机器人姿态角和移动速度控制误差较小, 能够有效提高移动机器人视觉伺服控制精度。

#### 3.6.2 控制效果

在此基础上, 进一步进行移动机器人的碰撞检测, 通过公式 (23) 的计算得出碰撞次数的测试结果, 如表 2 所示。

根据表 2 中的数据可知, 在静态障碍物场景和动态障碍物场景下, 与文献 [2] 系统和文献 [3] 系统相比, 所设计系统的移动机器人碰撞次数较少, 表明所设计系统具有较好的移动机器人视觉伺服控制效果。

(下转第 189 页)