

Perception and Control Strategies for Driving Utility Vehicles with a Humanoid Robot

Christopher Rasmussen¹, Kiwon Sohn², Qiaosong Wang¹, and Paul Oh²
机器人识别它所乘坐的车辆，根据仪表板进行自我

定位，并自动对准以便与方向盘和油门踏板交互

Abstract— This paper describes the hardware and software components of a general-purpose humanoid robot system for autonomously driving several different types of utility vehicles. The robot recognizes which vehicle it is in, localizes itself with respect to the dashboard, and self-aligns in order to interface with the steering wheel and accelerator pedal. Low- and higher-level methods are presented for speed control, environment perception, and trajectory planning and following suitable for operation in planar areas with discrete obstacles as well as along road-like paths.

I. INTRODUCTION

As part of the recently concluded DARPA Robotics Challenge (DRC) trials [1], contestant robots needed to carry out a number of navigation and manipulation tasks. These tasks were meant to represent a set of skills sufficient for a robot to move from the edge of a disaster zone such as a damaged nuclear power plant to its interior, where it could assess and possibly repair critical systems. The *Vehicle* stage of the challenge [2] called for the robot to drive a golf-cart-like utility vehicle around obstacles to a target location, get out, and walk away (aka *egress*).

In this paper we present techniques for autonomously driving several different utility vehicles using a humanoid robot (the *DRC-Hubo* robot and specific vehicles are described in Sec. II). These methods were developed using a robot entered in the 2013 DARPA DRC trials, but this is *not* a description of our approach to the *Vehicle* task there. At the competition, we used a pure tele-operation approach tuned specifically for the course and low-bandwidth conditions described in the rules [3]. Here we describe a more general set of *autonomous* skills for driving such vehicles in a variety of static environments.

Autonomously-driven vehicles of course have a long history [4], [5], [6], [7], [8] with prominent milestones at the 2005 DARPA Grand Challenge (DGC) [9] and 2007 DARPA Urban Challenge (DUC) [10], [11]. Since the DUC, much progress in the field has come in the industrial sector as automobile manufacturers and Google have extensively refined and tested *driverless car* technologies [12] and in some cases begun to offer them as safety options on production vehicles. These vehicles are effectively robots, but there are a number of significant differences between them and humanoid robots with respect to the structure and difficulty of the driving task.

First, when driverless car technologies are embedded in full-size vehicles, weight, size, and power limitations on



Fig. 1. DRC-Hubo in Polaris vehicle. (Top) Close-up during indoor testing; (bottom) At 2013 DARPA DRC trials (image courtesy of IEEE Spectrum Magazine)

资源限制：人形机器人需要考虑重量、尺寸、功耗限

制，还要保持行走平衡，传感器和计算资源预算有限 the sensors and computers used are not severe. In contrast, the budget for all of these categories is quite limited on a battery-powered humanoid which must worry about balance for walking and maintaining adequate current to all of its joint motors. Moreover, sensor and robot geometry choices cannot be made solely to optimize driving performance, as the point of a humanoid robot is *versatility*. All design decisions impacting driving must be considered jointly with other critical tasks, which for the DRC included walking, stair/ladder climbing, and power tool and door handle manipulation.

Second, achieving adequate sensor coverage (i.e., “blind spot” elimination) on integrated driverless car systems is generally just a matter of arraying enough fixed camera/sonar/lidar/radar units around the vehicle periphery and on its roof. Conversely, many parts of the road scene are inherently occluded from a humanoid robot inside a vehicle even with omnidirectional sensors on its head. And since the weight/space budget mentioned above makes fewer sensors more desirable and therefore less complete coverage

传感器覆盖：人形机器人坐在车内，视野受限，无法

*This work was supported by DARPA award #N65236-12-1-1005

¹Dept. Computer & Information Sciences, University of Delaware, USA.
E-mail: cer@cis.udel.edu

²Dept. Mechanical Engineering, Drexel University, USA

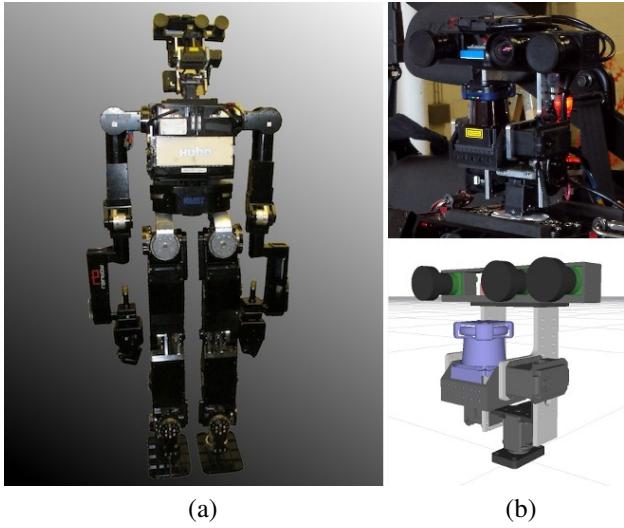


Fig. 2. (a) DRC-Hubo, (b) Sensor head photo detail and CAD model in ROS visualization [18].

时间来设置和校准，并且没有其他移动、纵或感知能力 unavoidable, the robot must carry out *view planning* to decide where to “look” depending on what it is doing from moment to moment.

必须物理操作方向盘和踏板，而不是电子信号控制，存在对齐、稳定和滑动等机械问题

Finally, the current generation of driverless cars are typically drive-by-wire, making motion control (steering, acceleration/braking, and shifting) and vehicle state feedback (speed, steering angle, engine temperature, and so on) trivial to implement and basically error-proof. Although there are specialized machines for actuating steering, acceleration, braking, and gear shifting from the driver’s seat with no permanent vehicle modifications [13], [14], [15], these take considerable time to set up and calibrate and have no other mobility, manipulation, or perception abilities. On the other hand, a humanoid robot manipulating the steering wheel and pedals is a mechanical system that must self-align, self-stabilize, and monitor for slips and other mishaps. Furthermore, vehicle state variables are not accessible by the robot through simple function calls; rather, they must either be visually read from the dashboard display or inferred from the robot’s own sensors and transformed into the vehicle frame.

必须从仪表板显示屏上直观地读取，或者从机器人自己的传感器中推断出来，并转化为车辆框架

The main contribution of this paper is a demonstration of the feasibility of a general-purpose humanoid robot driving an unmodified vehicle. The only previous work we can find on humanoid robot vehicle handling is [16], [17], in which an HRP-1 drove a modified forklift and a backhoe, but all control was via tele-operation using a video feed. Here we present a set of perceptual and physical methods which are sufficient to (1) *interface* the robot with different vehicles such that it can reliably accelerate, stop, and steer them as commanded; and (2) perform simple sensing and motion planning while *driving* given its limited and often occluded views.

II. EQUIPMENT

A. DRC-Hubo

Our robot, based on the earlier-generation KAIST Hubo 2+ [19], is pictured in Fig. 2(a). It is 1.40 m tall with a



Fig. 3. Vehicles driven: (a) Club Car DS and interior detail; (b) Polaris Ranger XP900 and interior detail. The windshield on the Club Car and the side netting on the Polaris were not present for the experiments reported here. The loose wires in the interior images are part of wireless e-stops added to the vehicles.

wingspan of 2.04 m, weighs 60 kg, and has $N = 33$ degrees of freedom (DoF): 1 in the waist, 6 per leg, 7 per arm, 1 in the left hand fingers, 2 in the right hand fingers, and 3 in the neck/sensor head. Three fingers on each hand close together via one motor for power grasps, and on the right hand there is an additional “trigger” finger which moves independently. Each hand also has a *peg* opposite the palm/fingers side (short versions are shown in Fig. 2(a) and longer ones in Fig. 5) which can be used as a point contact when the robot is in a quadrupedal walking mode. We have also found them useful as essentially rigid fingers for gross manipulation tasks such as turning the steering wheel, explained in Sec. IV-A.

The sensor head on the robot, shown in Figs. 2(a) and (b), was designed and built by us. It pans $\pm 180^\circ$ and tilts $\pm 60^\circ$ without self-collision, and has the following sensors which are relevant to this work:

- **3 × Pt. Grey Flea3 cameras**, each with about $90^\circ \times 70^\circ$ field of view (FOV), forming a synchronized stereo rig with baselines of 6 cm, 12 cm, and 18 cm.
- **Hokuyo UTM-30LX-EW laser range-finder** which scans at 40 Hz over a 270° FOV at an angular resolution of 0.25° . The minimum detectable depth is 0.1 m and the maximum is 30 m, and intensity-like reflectance information is provided for each point. The Hokuyo is mounted on a dedicated *tilting* servo which has a range of $\pm 60^\circ$ for point cloud capture
- **Microstrain 3DM-GX3-45 IMU** with 3-axis accelerometer, 3-axis gyro, and GPS receiver/antenna

B. Vehicles

Two different utility vehicles, shown in Fig. 3, comprise the set of *known* vehicles \mathcal{V} used for this work: an electric Club Car DS and a gas-powered Polaris Ranger XP900. Common features of these vehicles which distinguish them from passenger cars are an open cabin with the roof

测试使用两种实用车辆：

- Club Car DS电动车
- Polaris Ranger XP900燃油车

两车共同特点：开放式驾驶室，屋顶由细柱支撑，前排为长凳式座椅，无中控台。主要区别是Polaris更大，地板有传动轴凸起。



Fig. 4. Orthographic projection of major coordinate frames (vehicle shown is CAD model of Polaris from Gazebo simulator). The `/sensor_head` frame moves with the robot. Throughout this paper we use the ROS [18] convention for coordinates of $+X$ forward, $+Y$ left, and $+Z$ up.

supported by relatively thin *pillars*; and a bench-like front seat and no center console, making movement between the passenger's and driver's side during ingress/egress possible. However, there are some key geometric differences. The Polaris is larger overall, and it has a bump on the floor covering the drive shaft, visible in Fig. 3(b). Driving disparities (different steering ratios, implications of electric motor vs. gas engine, etc.) are discussed in Sec. IV-C.

定义了车辆内部坐标系，原点位于前排座

III. INTERFACING

椅前沿、地板和车辆中心线的交点。

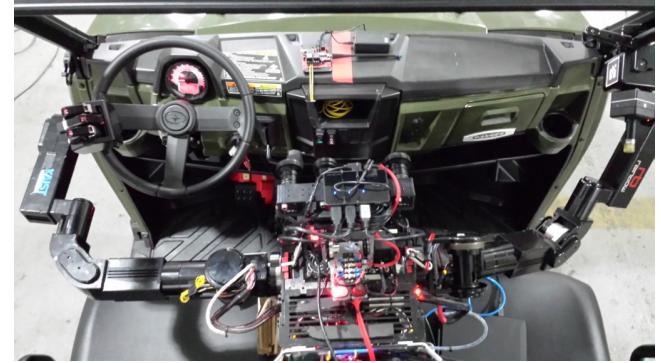
In the 2013 DRC trials *Vehicle* task [2], humans were allowed to set the robot up inside a vehicle in a *drive-ready* posture \mathbf{X}_{drive} rather than having it attempt to approach the vehicle, step up, and seat itself (aka *ingress*). By *posture* we mean the complete robot state which combines its joint state $\Theta = [\theta_1, \theta_2, \dots, \theta_N]$ and its pose $\mathbf{P} = [x, y, z, \alpha, \beta, \gamma]$ in the `vehicle_interior` frame.¹ For DRC-Hubo this posture, an example of which is pictured in Fig. 5(a), consists of: (a) the robot's torso offset to the right of the steering wheel; (b) its left hand peg inserted between the steering wheel "spokes"; (c) its right hand resting on its lap or grasping a vehicle support structure; and (d) its left foot near the accelerator pedal with its right foot flat on the floor.

This general posture works well for a variety of different utility vehicles, but it must be parametrized by the geometry of each vehicle in terms of steering wheel height off the floor, tilt, radius, and spoke arrangement; roof pillar spacing, angle, and cross-sectional shape/thickness; lateral location of accelerator; and so on. Thus, each *drive-ready* posture is specific to a vehicle v : $\mathbf{X}_{drive}^v = \{\Theta_{drive}^v, \mathbf{P}_{drive}^v\}$, and renderings of these are shown for each vehicle in \mathcal{V} in Fig. 5(b) and 5(c).

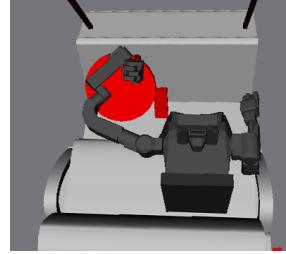
驾驶准备姿态

每个车辆都有特定的驾驶准备姿态 \mathbf{X}_{drive} ，包括：

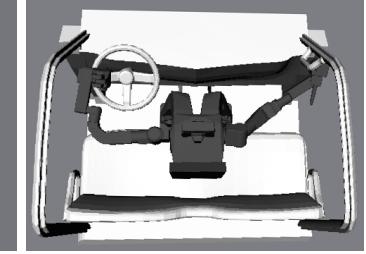
- 机器人躯干偏向方向盘右侧
- 左手钉插入方向盘辐条间
- 右手放在腿上或抓握车辆支撑结构
- 左脚靠近油门踏板，右脚平放地板



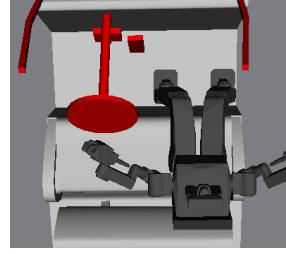
(a)



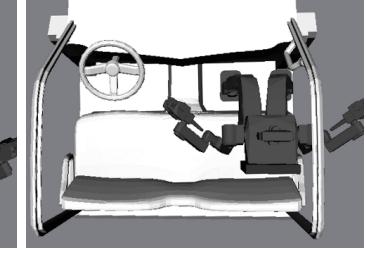
(b)



(c)



(d)



(e)

Fig. 5. (a) Top view of a drive-ready posture in Polaris vehicle; (b) $\mathbf{X}_{drive}^{clubcar}$ in OpenRAVE simulation [20]; (c) $\mathbf{X}_{drive}^{polaris}$ in OpenRAVE; (d, e) Neutral joint state $\Theta_{neutral}$ in Club Car and Polaris (exact pose is not required)

we assume that a high-resolution 3-D point cloud \mathcal{C}_{dash}^v of the *dashboard*² of each vehicle $v \in \mathcal{V}$, aligned with the `vehicle_interior` frame, is available as a *reference* for the robot. The dashboard reference clouds used here (shown from different views in Fig. 8 and Fig. 9) were acquired by the ladar on the sensor head tilting at $1^\circ/s$, voxelized to 0.025 m resolution, trimmed of all background features, and are *X**YZ* only.

Autonomous interfacing works as follows:

- The robot is placed (or arrives on its own) in the passenger seat in a neutral/vehicle-agnostic sitting position $\mathbf{X}_{neutral}$, illustrated in Fig. 5(d) and 5(e) for each vehicle. The passenger side is advantageous kinematically for steering and entering on that side avoids collision issues with the steering wheel.
- The robot obtains a 3-D point cloud/image capture

自主接口流程

- 机器人以中立姿态坐在乘客座位
- 获取仪表板区域3D点云以识别车辆类型
- 通过点云配准精确估计在车内的位置
- 制定并执行“蹭行”计划移动到驾驶位置

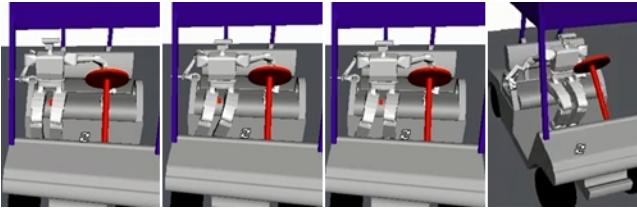


Fig. 6. One cycle of *scooting* motion as robot moves from neutral to drive-ready position, simulated for the Club Car in OpenRAVE

- $\mathcal{C}_{neutral}$ of the rough dashboard area in front of it in order to infer the vehicle \bar{v} it is in.
- (3) It precisely estimates its initial pose within the vehicle $\mathbf{P}_{neutral}^{\bar{v}}$ by registering $\mathcal{C}_{neutral}$ to the reference $\mathcal{C}_{dash}^{\bar{v}}$.
 - (4) It formulates and executes an *interfacing* plan to move to $\mathbf{X}_{drive}^{\bar{v}}$. The leftward movement on the seat to reach the drive-ready posture is a stereotyped quadrupedal motion which we call “scooting”. After each motion cycle, one of which is shown in Fig. 6, a new point cloud is scanned and step (3) is repeated to decide how far to move or to stop.

A. Vehicle recognition

A. 车辆识别

利用两车仪表板宽度差异 (Polaris宽约0.35米) 进行识别：

- 对点云做垂直平面拟合
- 计算内点间最大横向距离
- 通过阈值判断车辆类型

the error after performing a fit to each reference cloud is not a reliable match indicator. Instead, we exploit the size discrepancy between the vehicles to discriminate them. In particular, the Polaris dashboard is about 0.35 m wider, so after doing a robust vertical plane fit to $\mathcal{C}_{neutral}$, the maximum lateral distance between inliers can be thresholded, following techniques we described in [25], to infer \bar{v} . An alternative method to estimate the width is to first detect the roof pillars, which form parallel, nearly vertical cylinders in $\mathcal{C}_{neutral}$. In Sec. III-B we do the equivalent with a single level radar scan.

B. 车内定位

提供两种方法估计机器人相对车辆的位置：

1. 完整方法：使用RANSAC鲁棒配准算法，配合Fast Point Feature Histograms (FPFH)特征匹配和迭代最近点(ICP)精化
2. 快速方法：使用单次水平激光扫描识别车顶支柱，提取偏航角和xy位置

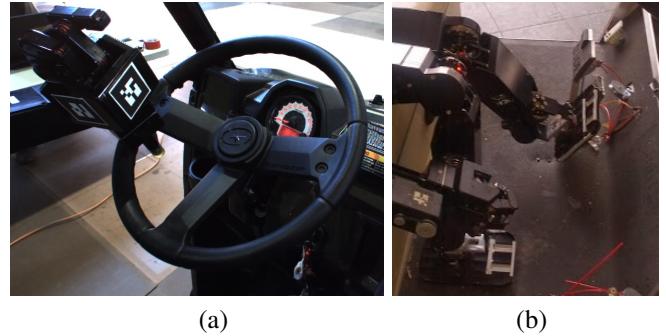


Fig. 7. (a) Peg-in-wheel method for steering (Polaris); (b) Foot arrangement for pedal control (Club Car)

point correspondences, *feature signatures* are calculated to aid matching. We use the Point Cloud Library’s (PCL) [27] Sample Consensus Initial Alignment with Fast Point Feature Histograms (FPFH) [28] to perform this step. The initial aligning transform is then refined using Iterative Closest Points (ICP) [29].

Another, much faster but only partial technique for estimating the sensor head pose works with a single approximately level radar scan. Assuming the robot head is above the height of the dashboard, this scan slices through the vehicle roof pillars, which form two tight clusters. After removing points more than a few meters away, there may still be some objects inside the vehicle (such as the steering wheel or the robot’s own hands), so the robot does Euclidean cluster extraction [27] with a small maximum cluster size. All cluster pairs are then checked for 2-D geometric feasibility (distance, angle, etc.), and the most likely feasible pair is used to extract the sensor head yaw and t_x, t_y .

IV. DRIVING

We make several important assumptions about the state of the vehicle when the robot begins interfacing: (1) It is powered on or the engine is running; (2) It is in forward mode (Club Car) or drive gear (Polaris); (3) The tires are straight and therefore the steering wheel’s orientation is known.

The drive-ready positions depicted in Fig. 5 for the Club Car and Polaris show the robot’s right hand resting on its

基本假设

- 车辆已启动或发动机运行
- 处于前进档位
- 轮胎笔直，方向盘方向已知

A. Steering actuation and sensing

The robot turns the steering wheel by *dialing*: it moves its left hand in a circular trajectory with the peg inserted between the steering wheel spokes as shown in Fig 7(a). This motion was chosen over holding the wheel in the traditional manner because it has no singularities, avoiding the necessity of regrasping when turning through large angles. The lack of a slip ring in DRC-Hubo’s wrist roll joint precludes grasping

A. 转向驱动和感知

采用“拨号”方法转向：

- 左手钉插入方向盘辐条间做圆周运动
- 避免了传统握持方式的奇异点问题
- 最大转向速度120°/s

转向控制的关键限制：

- 间隙问题：只有“推”辐条时才能转向，换向时有延迟
- Club Car间隙约60°, Polaris约90°
- 无法抵抗外力（如坡度、颠簸）

hand motion only causes steering wheel motion when it is “pushing” one of the spokes—when the steering direction is changed, there is a gap that the peg must cross before engaging another spoke. As can be seen in Fig. 3, the smallest (depending on which pair of spokes the peg is inserted between) backlash angle β_v for vehicle v is about 90° for the Polaris and 60° for the Club Car. Consequences of the backlash are: (1) Additional bookkeeping in the steering wheel controller, as the hand angle is not necessarily the same as the steering wheel angle (aka $\phi_{hand} \neq \phi_{steer}$); (2) A delay to change the sign of the steering rate $\dot{\phi}_{steer}$ of at least $\tau_{\beta_v} = \dot{\phi}_{hand}/\beta_v$ s; and (3) Inability to resist external forces on the tires due to slope, bumps, etc. which might tend to amplify the turning rate.

DRC-Hubo’s left arm joint encoder values and forward kinematics are used to derive ϕ_{hand} , which with a known initial steering wheel angle yields ϕ_{steer} . This can be refined somewhat by sensing contact between the peg and spoke using a force-torque sensor in the wrist. Even with some misalignment, given the current peg length there is no danger of it slipping out of the steering wheel plane during motion.

B. 速度驱动和感知

- 仅通过左脚踝关节角度控制油门踏板
- 工作范围：踝关节角度-35°到5°
- 停车策略：松开油门让车辆滑行停止
- 使用立体视觉里程计估算当前车速

不使用刹车的原因：

- 当前姿态下左脚移动到刹车踏板会违反关节限制
- 双脚方案虽可行但稳定性差，转向不便，移动困难

In the current drive-ready position, the hip yaw required to translate the left foot over to the brake pedal would violate joint limits on DRC-Hubo. A two-footed approach (left on brake, right on accelerator) is possible, but would require a sitting position that is (a) less stable because of the narrower

C. 运动控制

速度控制：

简单的目标速度跟踪，根据速度误差调整踏板角度。

两种运动控制策略：

1. 停车转向模式：

- 只在停车时转向，消除转向延迟问题
- 遵循Dubins路径（圆弧和直线段组合）
- 每段执行：停车→转向→加速→行驶→停

车

2. 连续轨迹跟踪：

- 使用横向误差转向控制器
- 参考DARPA大挑战赛方法
- 根据轨迹误差和航向差计算转向角

$$\theta_{LAP}(t) = \begin{cases} \frac{\Delta \theta_{LAP}}{T} & \text{if } t < T \\ \theta_{LAP} & \text{otherwise} \end{cases} \quad (1)$$

where currently $T = 0.5$ m/s and $\Delta \theta_{LAP} = 1^\circ$.

There are currently two strategies for overall motion control of the vehicle, detailed below.

a) *Stop-to-steer*: To remove the concerns from Sec. IV-A about the limited turning rate achievable with $|\dot{\phi}_{hand}|_{max}$ and the backlash lag τ_{β_v} on direction changes, the robot is constrained to only turn the steering wheel while stopped. This results in the vehicle following a Dubins path consisting of circular arc and straight line segments, which may be the original form of the motion plan or simply an approximation of a more complicated trajectory, subject to a full stop between each segment.

Suppose the current step i of an M -step plan (see Sec. IV-D) calls for the stopped vehicle to traverse a segment with curvature κ_i and arc length d_i . With Ackermann steering, the steering ratio r_v , wheelbase w_v , and steering wheel backlash β_v of the vehicle imply a steering *hand target angle* $\phi_{hand}^{target} = f(\kappa_i, r_v, w_v, \beta_v)$, which is then executed. For high κ , this may take several seconds to complete.

ϕ_{hand}^{target} is then set (1 m/s is the current default), and the vehicle speed controller will slowly push down the pedal until the vehicle begins to move and reaches its target speed. As the vehicle moves, the forward motion component $\tilde{v}_x(t)$ is integrated until d_i is exceeded, at which point v_x^{target} is set to 0 and the next plan step $i+1$ is obtained. If at any time during motion an imminent collision is detected because of dynamic obstacles or deviation from the planned trajectory, motion is also halted.

b) *Continuous trajectory following*: In this mode the robot attempts to follow an arbitrary continuous trajectory without stopping completely. We use a version of the *cross-track error* steering controller from [9]. Modifying their notation slightly to fit ours:

$$\phi_{tires}(t) = \Psi(t) + \arctan \frac{ky(t)}{v_x(t)} \quad (2)$$

D. 高级感知和路径规划

环境感知：

- 使用激光雷达扫描检测障碍物
- 建立3D占用网格地图 (OctoMap)
- 投影为2D代价地图用于路径规划

路径规划：

- 使用搜索式规划库(SBPL), 考虑阿克曼运动约束
- 支持人工提供高级路径段
- 实现道路跟踪功能, 使用粒子滤波优化路径假设

When not moving, the robot can obtain a point cloud of the scene outside the vehicle with a single ladar sweep and detect obstacles as outliers to a planar ground fit.

For more efficient storage and processing, ladar scan points from the point cloud are inserted into a 3-D occupancy grid—here we use OctoMap [34] with a minimum cell size of 0.2 m. The map is *rolling*—we are only concerned with what is immediately in front of and around the robot, rather than trying to build a full map of the environment it is traveling through—so the occupancy grid is roughly 10 m on a side, centered forward of the vehicle driving origin.

Each scan point can be annotated with relevant information such as its intensity or obstacle/free classification before insertion into the occupancy grid. In the latter case, this leads to aggregation of multiple observations such that we can get the freespace likelihood for each grid square. The motion estimates furnished by the visual odometry module are crucial in allowing the robot to reason about obstacles it observed previously but can no longer see.

Projecting the current 3-D occupancy grid down to a 2-D costmap allows the robot to generate a feasible trajectory to a goal pose by calling a path planner such as the Search-Based Planning Library (SBPL) [35], [36] which is aware of the vehicle's Ackermann motion constraints. Human operators may also provide high-level plans to the robot by inspecting the current obstacle costmap and supplying Dubins-like segments to be executed via the stop-to-steer method of Sec. IV-C.

A final perception and planning mode we have developed is for path- or road-following. Following our methods for trail-following detailed in [37], [38], we can track a path in front of the vehicle by formulating a *path likelihood* function which measures how well a low-dimensional path shape hypothesis (width, lateral offset, relative heading, and curvature) agrees with sensor measurements and continuously optimizing it via particle filtering. The likelihood function used here simply measures the proportion of ground fit inliers to outliers taken from the costmap introduced above, although more sophisticated appearance cues could easily be incorporated. The centerline of the tracked path supplies the cross-track error and path segment orientation needed for the continuous trajectory following method in Sec. IV-C.

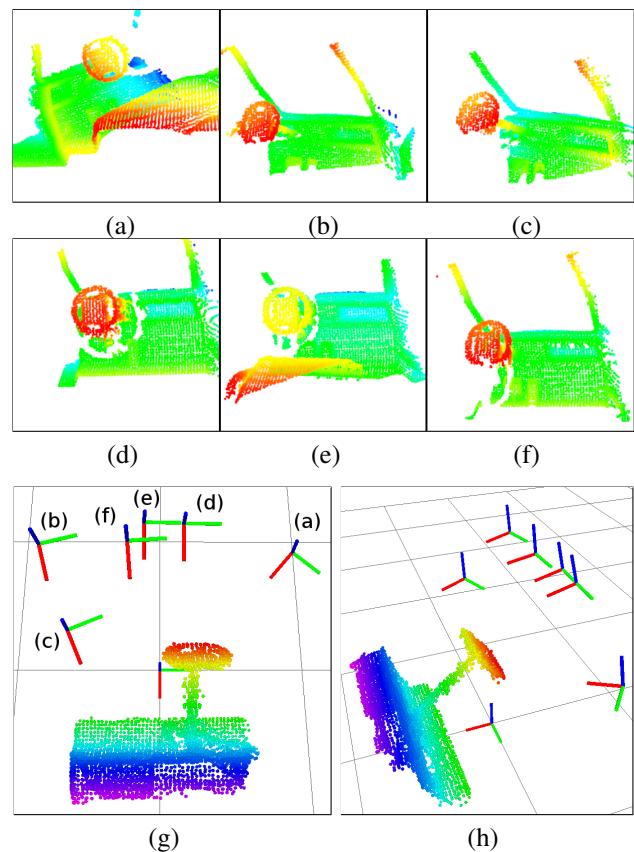


Fig. 8. Club Car dashboard registration for sensor head pose estimation. (a-f) Hokuyo point clouds acquired from different poses, in sensor_head frame; (g, h) Views of estimated sensor head locations with respect to the dashboard reference cloud and origin (all point clouds colored by X, grid squares are 1 m)

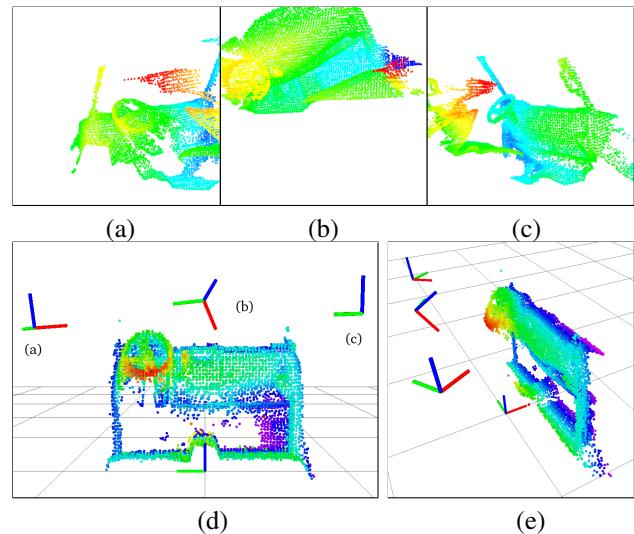


Fig. 9. Polaris dashboard registration for sensor head pose estimation. (a-c) Hokuyo point clouds; (d, e) Views of estimated sensor head poses with respect to reference cloud

V. RESULTS

Dashboard point cloud registration and sensor head pose estimation results are shown for the Club Car in Fig. 8 and

仪表板配准结果

- 在两种车辆中从约10个不同位置采集数据
- 平均配准时间约5秒 (Intel Core i7-3930K处理器)
- 除一个案例外都获得很好结果 (唯一异常有10°横滚误差)

驾驶测试

- 广泛测试了所有转向和速度控制方法
- 在室内外环境演示了停车转向运动控制
- 集成仿真验证了障碍检测、建图、规划和轨迹跟踪
- 收集了5公里以上的校园和高尔夫球场数据进行算法验证

具体演示

- 室内障碍区域成功穿越序列
- Gazebo仿真中的占用网格和轨迹规划
- 高尔夫球场场景的三点掉头机动，展示了地面/障碍分割和高精度着色的占用地图

environments. Several versions of the stop-to-steer motion control method from Sec. IV-C were demonstrated in the Polaris in both indoor and outdoor environments, shown in Fig. 1. A sequence from one of many indoor obstacle avoidance tests with DRC-Hubo at the wheel, using human-provided plan segments, is shown in Fig. 10.

The obstacle detection, mapping, search-based planning, and continuous trajectory following methods from Sec. IV-D have been demonstrated in an integrated fashion in simulation (using ground-truth odometry). A snapshot of a Gazebo simulation is shown in Fig. 11(a), with the generated Octomap and planned trajectory several frames later shown in Fig. 11(b).

The perception and path tracking algorithms have further been validated offline using data collected with the sensor head while driving manually 5+ km around campus and golf course testing areas (samples pictured in Fig. 11(c) and (d)). Steps of the pipeline from an area where the vehicle was driven through a 3-point turn are represented in Fig. 11(e). The two left images show the scene and its corresponding point cloud, while the two right images show the ground/obstacle segmentation and height-colored Octomap of the point cloud, with the visual odometry-derived motion estimate of the maneuver overlaid.

VI. CONCLUSION

This work is a preliminary demonstration of the feasibility of humanoid robots driving vehicles, and much remains to be done to make the system more practical, robust, and general. Beyond more integrated live testing of continuous trajectory following and path tracking, of immediate interest would be adding skills to relax the assumptions in Sec. IV, including turning the vehicle on/off with a key or switch and gear shifting to allow reverse maneuvers in the motion planner. Both of these would require more dexterous manipulation, including force feedback, and visual analysis for hand-eye coordination.

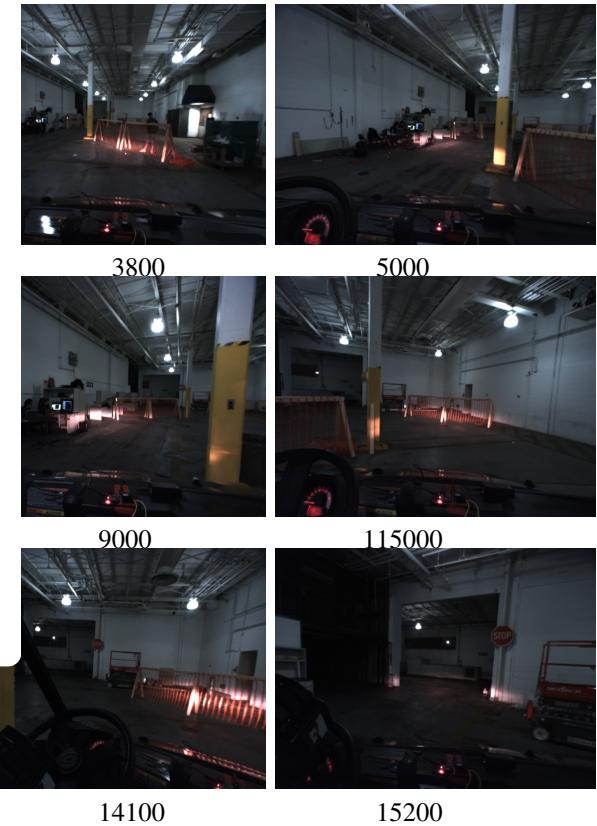


Fig. 10. Image sequence (with frame numbers) from sample successful Polaris traversal of garage obstacle zone. Stop-to-steer tele-operation was used.

未来工作方向

即时改进：

- 集成连续轨迹跟踪和路径追踪的实时测试
- 放宽基本假设：添加车辆启动/关闭、换挡等技能
- 需要更灵巧的操作和力反馈

长期扩展：

- 使用视觉分割检测车道线、标志等
- 处理动态环境中的其他车辆
- 解决视野规划问题，包括躯干倾斜以减少盲点
- 扩展到更大的车辆集合
- 添加自主上下车功能
- 读取视觉指示器如速度表和档位状态

这项研究为人形机器人在灾难救援、危险环境作业等场景中的应用开辟了新的可能性。

[3] —, “DARPA Robotics Challenge DRC Trials Rules (Release 7),” 2013, available at <http://www.theroboticschallenge.org/files/DRCTrialsRulesRelease7DISTAR22157.pdf>. Accessed December, 2013.

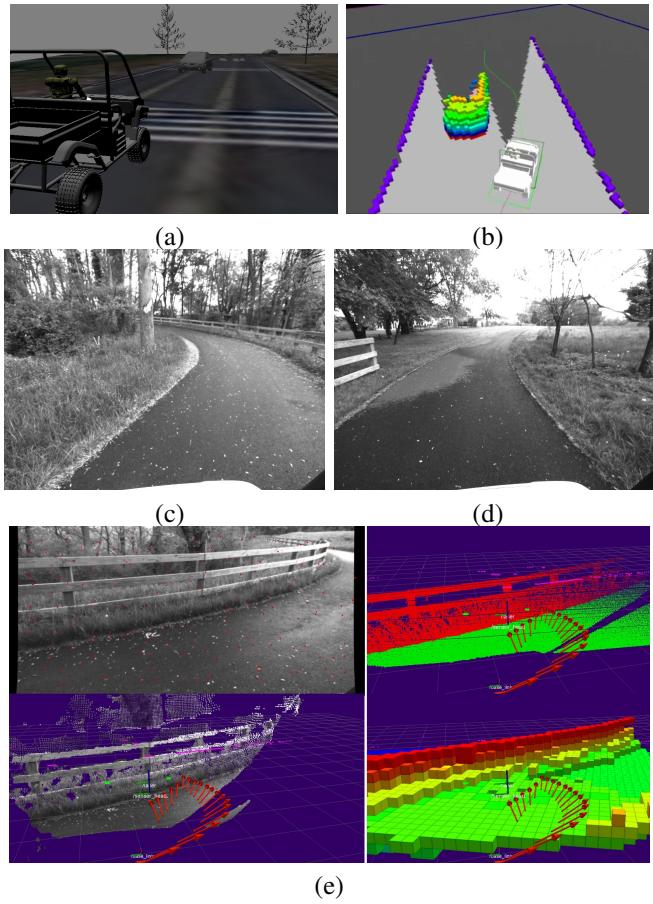


Fig. 11. (a) ROS Gazebo driving simulation; (b) Corresponding *rviz* visualization of occupancy grid, plan generation, and trajectory following; (c, d) Sample terrain in golf course data collection area; (e) Upper-left: golf course scene; lower-left: dense stereo reconstruction with motion history overlaid (vehicle stopped and reversed); upper-right: freespace/obstacle segmentation after ground plane fit; lower-right: Octomap occupancy grid colored by height

- [4] J. Crisman and C. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2496–2501.
- [5] E. Dickmanns and B. Mysliwetz, "Recursive 3-d road and relative ego-state recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 14, pp. 199–213, 1992.
- [6] D. Pomerleau, "RALPH: Rapidly adapting lateral position handler," in *Proc. IEEE Intelligent Vehicles Symposium*, 1995, pp. 506–511.
- [7] B. Southall and C. Taylor, "Stochastic road shape estimation," in *Proc. Int. Conf. Computer Vision*, 2001, pp. 205–212.
- [8] N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intelligent Vehicles Symposium*, 2003.
- [9] S. Thrun, M. Montemerlo, et al., "Stanley, the robot that won the DARPA grand challenge," *J. Field Robotics*, vol. 23, no. 9, 2006.
- [10] C. Urmson, et al., "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robotics*, vol. 25, no. 1, 2008.
- [11] A. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Multi-sensor lane finding in urban road networks," in *Robotics: Science and Systems*, 2008.
- [12] B. Bilger, "Auto correct: Has the self-driving car at last arrived?" *The New Yorker*, November 2013.
- [13] S. Shoval, J. Zybtur, and D. Grimaudo, "Robot Driver for Guidance of Automatic Durability Road (ADR) Test Vehicles," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1998.
- [14] Stahle Robot Systems, "AUTONOMOUS DRIVING SYSTEM SFP2000FF for cars," 2014, available at <http://www.stahle.com>. Accessed January, 2014.

- [15] Kairos Autonomi, "Pronto4 Agnostic Autonomy System for Existing Vehicles or Vessels," 2014, available at http://www.kairosautonomi.com/pronto4_system.html. Accessed January, 2014.
- [16] H. Hasunuma, M. Kobayashi, H. Moriyama, T. Itoko, Y. Yanagihara, T. Ueno, K. Ohya, and K. Yokoi, "A Tele-operated Humanoid Robot Drives a Lift Truck," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002.
- [17] K. Yokoi, K. Nakashima, M. Kobayashi, H. Mihune, H. Hasunuma, Y. Yanagihara, T. Ueno, T. Gokyu, and K. Endou, "A Tele-operated Humanoid Robot Drives a Backhoe in the Open Air," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2003.
- [18] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: An Open-Source Robot Operating System," in *Proc. ICRA workshop on Open-Source Software*, 2009.
- [19] R. Ellenberg, R. Sherbert, P. Oh, A. Alspach, R. Gross, and J. Oh, "A common interface for humanoid simulation and hardware," in *Proc. IEEE Inter. Conf. on Humanoid Robots*, 2010.
- [20] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, August 2010.
- [21] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. European Conf. Computer Vision*, 2004.
- [22] J. Assfalg, M. Bertini, and A. D. Bimbo, "Content-based retrieval of 3-d objects using spin image signatures," *IEEE Trans. Multimedia*, vol. 9, no. 3, 2007.
- [23] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2010.
- [24] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *IEEE Inter. Conf. on Robotics and Biomimetics*, 2011.
- [25] C. Rasmussen, K. Yuvraj, R. Vallett, K. Sohn, and P. Oh, "Towards Functional Labeling of Utility Vehicle Point Clouds for Humanoid Driving," in *IEEE Inter. Conf. on Technologies for Practical Robot Applications*, 2013.
- [26] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [27] R. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Inter. Conf. on Robotics & Automation*, 2011.
- [28] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.
- [29] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [30] K. Sohn, "Optimization of humanoid's motions under multiple constraints in vehicle handling task," Ph.D. dissertation, Mechanical Engineering and Mechanics, Drexel University, May 2014.
- [31] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *J. Field Robotics*, vol. 23, no. 1, 2006.
- [32] A. Howard, "Visual odometry for motion estimation," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2008.
- [33] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D Reconstruction in Real-time," in *IEEE Intelligent Vehicles Symposium*, 2011.
- [34] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.
- [35] M. Likhachev. (2013) Ros search-base planning library stack. [Online]. Available: <http://www.ros.org/wiki/sbp/>
- [36] M. Pivtoraiko and A. Kelly, "Generating Near Minimal Spanning Control Sets for Constrained Motion Planning in Discrete State Spaces," in *Proc. IEEE/RSJ Inter. Conf. on Intelligent Robots & Systems*, 2005.
- [37] C. Rasmussen, Y. Lu, and M. Kocamaz, "Integrating stereo structure for omnidirectional trail following," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2011.
- [38] ———, "A trail-following robot which uses appearance and structural cues," in *Proc. Int. Conf. on Field and Service Robotics*, 2012.