Early Phases of Humanoid Vehicle Ingress Using Depth Cameras

Christopher Rasmussen[1], Kiwon Sohn[2], Karthikeyan Yuvraj[2], and Paul Oh[2]

***Abstract*— This paper presents work on integrating perception and motion-planning for a humanoid robot to *ingress*, or enter, a small utility vehicle as a precursor to driving, a stage in the recent DARPA Robotics Challenge (DRC). Using a Hubo 2+ robot platform and a pair of RGB-D cameras, we describe a set of approaches to and present results on the first four phases of ingress: (1) visually *search* for the vehicle's doorway as a target to walk toward, (2) plan and execute a collision-free *approach* to the doorway via walking using visual odometry, (3) make visually-guided fine positioning adjustments near the entry door during *docking*, and (4) *step* from the ground up to the floor of the vehicle. All recognition is done on 3-D point clouds derived from the depth cameras without appearance information. Some further ingress results from the DRC-Hubo making use of grasping are also shown.**

I. Introduction

As part of the recent DARPA Robotics Challenge (DRC) trials [1], contestant robots were challenged to carry out a series of navigation and manipulation tasks which would take them from the edge of an accident zone at a location such as a nuclear power plant to an area where they can survey and possibly repair damage. The original definition of the first stage in the challenge tasked the robot to approach a car-like vehicle, enter it, drive it to a target location, get out, and walk to a building. We name these stages *ingress*, *driving* proper, and *egress*, respectively.

There are a large number of perceptual, motion planning, and control problems to solve in order to successfully complete the driving challenge. In this paper we focus only on the *ingress* stage. which we break into several phases. Consider Fig. 1. From its initial position, the robot must first *search* for the vehicle it will drive in a possibly cluttered scene. Not just any part of the vehicle will do–the robot must also locate and choose one of its doors as the planned entry point. Next the robot must plan a walking path to its goal around possible obstacles and accurately follow that trajectory to *approach* the vehicle. When the robot gets close enough to the vehicle it must start to visually home on its target door in order to properly align itself or *dock* for the final phase of ingress, actually *stepping* up into the vehicle without colliding with it and while maintaining balance.

Subsequent stages of the ingress task which will not be examined here include executing the *sitting* maneuver, an analogous step to docking while seated to get precisely aligned with the steering wheel which we call *scooting*, and finally *interfacing*. During interfacing all control surfaces of interest such as the vehicle on/off switch, steering wheel, accelerator/brake pedals, and the gear shifter (for reversing) must all be located and parametrized. Furthermore, the robot must carry out a set of calibrations before the vehicle begins moving such as checking the reachability of these control surfaces, grasping/touching them, inferring or refining expected affordances, measuring force resistance, etc.

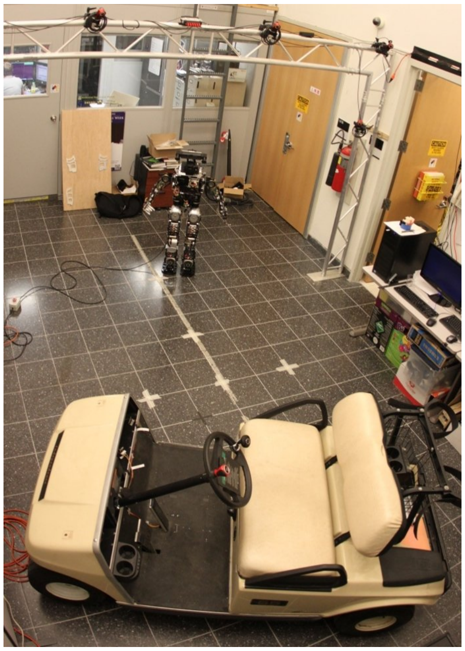


Fig. 1. Hubo 2+ robot in testing area at beginning of *search* phase of ingress.

A. Related Work

There has been much recent work on 3-D object detection using depth cameras and similar sensors [2], [3], [4], [5]. Additionally, [6] uses visual appearance, local shape and geometry, and geometric context features to label colored point clouds of indoor office and home scenes with a large number of classes such as *wall*, *keyboard*, *chairback*, *monitor*, *book*, and so on. [7] looks for objects such as cups, bowls, cereal boxes, etc. in point clouds with color information using an RGB-D variant of HOG detectors after first training on 3-D models. Similarly, much work has been done with the PR2 robot in terms of looking at tabletops and segmenting and identifying objects with its ladar and/or stereo cameras [8], including plane fitting and region growing for segmentation in a kitchen environment [9]. Also relevant is the work in [10] on door handle detection using a tilting Hokuyo after first finding doors using depth and reflectance information.

[1] Dept. Computer & Information Sciences, University of Delaware, Newark, DE 19716. E-mail: cer@cis.udel.edu

[2] Dept. Mechanical Engineering, Drexel University, Philadelphia, PA 19104

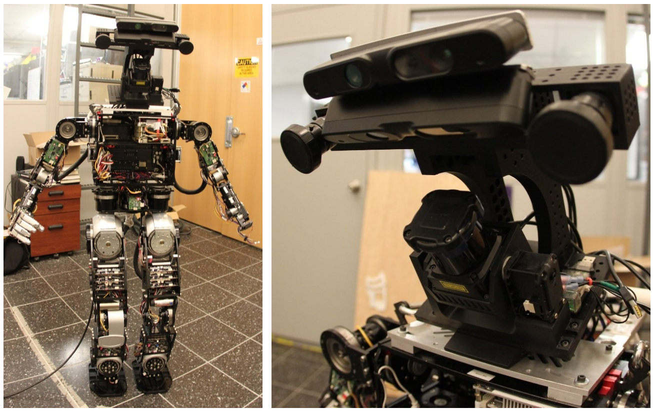


Fig. 2. (Left) Hubo 2+ robot in testing area with prototype sensor head; (right) Sensor head with two Asus Xtion Pro Live depth cameras. Stereo color cameras, a tilt Hokuyo, and an IMU are also part of the head but not used here.

There has been a substantial amount of work on image-based car recognition/detection in street scenes including [11], [12] and [13], which takes a bag-of-words approach, but none specifically that we could find which focuses on 3-D data.

Closely related work includes bipedal walking and navigation [14], [15], and efforts to get bipedal robots to step up or climb stairs. The Nao robot climbs a spiral staircase in [16] after stopping to acquire a point cloud with a short-range tilting Hokuyo and segmenting individual tread rectangles. The authors mention [17], [18] as key prior work on using RANSAC-like techniques to perceive a single step. The Honda Asimo robot sees isolated planar steps with a tilt ladar, stepping around or onto them in [19], [20]. There is also analysis of point clouds for stair perception in [21]. However, biped walking and stair climbing motions alone are not sufficient for the step phase of the ingress task, as cyclic gaits are not well suited to issues such as planning of end effector trajectories in cluttered environments while simultaneously considering balance.

[22] showed that the HRP-2 robot can step into a utility vehicle in a simulation environment. However, their approach made assumptions about gripping power which may not be achievable with current robots. [23] also demonstrated vehicle ingress motions using a humanoid model in a simulation toolkit. Their trajectory had a height requirement that the robot hip start above the seat, which is not always possible for such vehicles.

II. Equipment

For most of this work, our robot is a humanoid called Hubo 2+ [24] which is 130 cm tall and has a mass of 42 Kg. It has 38 total degrees of freedom (DoF): 6 DoF in each limb, 3 DoF in the neck, 1 DoF at the waist, and 5 DoF per hand. Normal walking speed is 0.5 m/s, with a maximum of 1.0 m/s. We also present some results with the DRC-Hubo, a larger and stronger successor to Hubo 2+, shown in Fig. 11.

A prototype sensor head, pictured on the right in Fig. 2, was used for the Hubo 2+ results here. The relevant sensors are two Asus Xtion Pro Live RGB-D cameras mounted with approximately a $45°$ pitch difference between them. Depth and color images were captured from each Asus at 320×240 resolution at 30 fps. We do not currently have an actuated neck strong enough for the sensor head, so it is mounted at a fixed pitch angle of $30°$ below horizontal. The combined vertical field of view of the cameras is about $80°$ which allows a view of the ground from the robot's feet out to the vehicle while walking, as can be seen in Fig. 3. The horizontal field of view is $60°$. When a wider view is required the robot pans its waist joint as pictured in Fig. 8. In contrast, the DRC-Hubo's sensor head can pan and tilt and has a Hokuyo ladar range-finder on a dedicated tilt servo for point cloud acquisition.

Outdoor operation would preclude the use of the RGB-D cameras shown here, so the sensor head prototype has other sensors which can provide depth information in sunlight. Although the use of intensity or color information in conjunction with the depth information from the cameras could certainly help our algorithms, we have chosen to focus exclusively on XYZ point clouds alone in order to make fusion of and switching between different sensors easier.

III. Search phase

Throughout this paper we use the ROS [25] convention for coordinates of $+X$ forward, $+Y$ left, and $+Z$ up.

A prerequisite for virtually all steps in this paper is the maintenance of a good ground plane estimate at all times, given that the cameras are pitched downward sharply and the robot sways back and forth as it walks. Having such an estimate allows us to *rectify* the entire point cloud, put the ground plane at $Z = 0$, and remove ground plane points by thresholding $|z| \leq 0.05$ m. We initially detect and parametrize the ground with a robust plane fit on nearby depth camera points using RANSAC [26], and smooth subsequent estimates with a Kalman filter and gating to prevent height jumps near parallel planes such as the vehicle floor (see Sec. V). A sample ground-obstacle segmentation for the scene in Fig. 3 is shown at the left in Fig. 7.

With the ground plane and a coarse shape model, the first part of the *search* phase—aka vehicle detection in the scene point cloud—can be framed as a 2-D template matching problem as we discussed in [27]. The last part of the search phase is to localize the *entry pose* $\mathbf{P}_{entry}$, or where the robot wants to be before stepping into the vehicle. The entry pose serves as a goal pose for the walking motion planner in the *approach* phase detailed in Sec. IV.

The entry pose is derived relative to a detected vehicle part, namely one of its *doors*. For the test vehicle used for this paper's results and many vehicles in this category, there is no door *per se* but simply a gap between the seat and the dashboard which we will call the *doorway*. This area is visible in Fig. 4. In order to avoid collision complications due to the steering wheel during the step phase, we prefer the passenger-side doorway. With the current sensor view registered to the vehicle and non-vehicle points trimmed

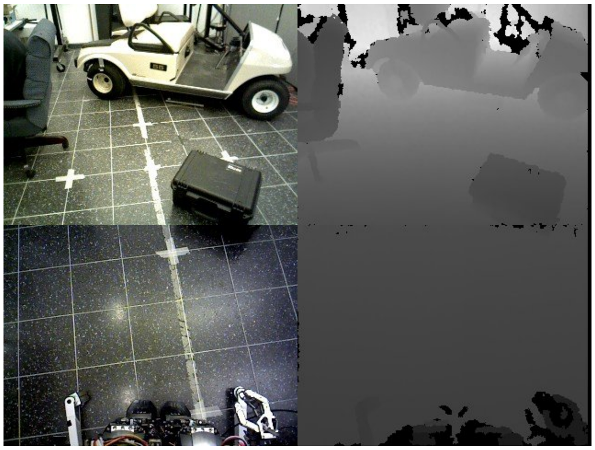
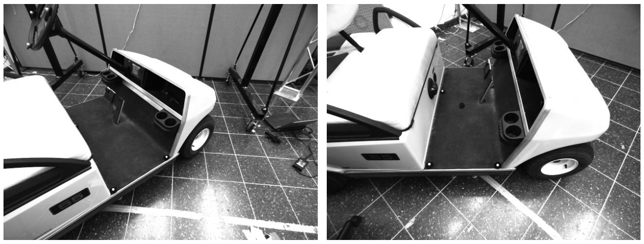
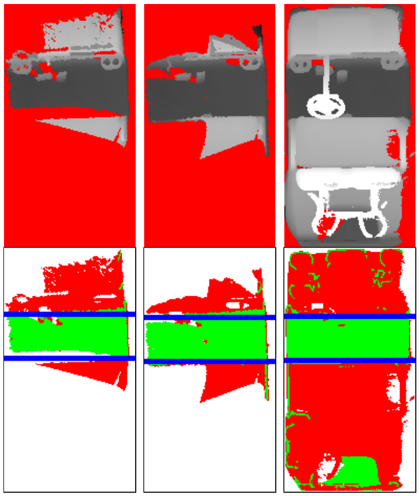


Fig. 3. Views from Hubo 2+ sensor head's pair of RGB and depth cameras. See Fig. 7 for associated ground/obstacle segmentations and sample motion planning results.



Passenger Rear (PR) Passenger Mid (PM)

Fig. 4. Images from sample capture locations near the vehicle (these are from the stereo camera in Fig. 2 and not used for perception here)



$\mathbf{F}^{PR}_{asus}$ $\mathbf{F}^{PM}_{asus}$ $\mathbf{F}_{veh}$

Fig. 5. (Top left) Registered and trimmed heightmaps of top Asus for scenes in Fig. 4; (top right) Vehicle heightmap $\mathbf{H}_{veh}$ derived from a KinFu (based on [28]) point cloud; (bottom) Inliers after horizontal plane fitting on floor height slice in green, outliers (all other vehicle points) in red, and fitted rectangles outlined in blue. "No data" points are white here.

away, the search for the doorway before walking commences is considerably constrained.

We find the doorway by looking for the *floor* beyond it, a planar rectangular region parallel to the ground plane whose nearest edge defines the doorway. To avoid confusion with the seat or hood planes we use our prior knowledge that the floor plane must be at a steppable height. Assuming that this is a roughly constant offset above the vehicle's ground clearance, we use specifications from the set of exemplar vehicles introduced in [27] to obtain minimum and maximum bounds on the floor height (in this case $z_{floor} \in [0.15, 0.40]$ m). Excluding points in the registered sensor point cloud outside of this range yields a nominal *floor slice* upon which we then run a RANSAC horizontal plane fit (i.e., the normal must be within $5°$ of vertical). This finds the floor plane in the $\mathbf{H}^*_s$ as well as $\mathbf{H}_{veh}$.

Fig. 5 shows the floor plane inliers for each view in green and outlier points from the vehicle in red. Isolating the cluster of inliers belonging to the floor region itself can be formulated as a constrained rectangle-finding problem. Assuming that the floor rectangle's axes are aligned with those of the vehicle, its width is approximately the same as the vehicle, and its center is on the vehicle centerline. This leaves only 2 free variables to determine $\mathbf{R}_{floor}$: (x_{floor}, l_{floor}), the floor's forward/backward position and dimension (i.e., the distance between the seat and the dashboard). We put reasonable bounds on these variables and again run a particle filter with the likelihood function $P_{floor}(\mathbf{R}) = (N_{in} - N_{out})/A$, where N_{in} is the number of floor plane inliers in $\mathbf{R}$, N_{out} is the number of outliers in the rectangle, and A is its area.

The search is very fast to converge and only 10 iterations are needed. The blue lines in Fig. 5 indicate the estimated floor rectangles. These are accurate when the whole floor region is visible, and conservative when it is not. Impingements like the pedals, drink holders, and steering wheel are detectable as outlier points present inside the floor rectangle, and this information can be passed on to the motion planner.

IV. Approach Phase

The approach phase begins when $\mathbf{P}_{entry}$ has been selected. Motion goals during this phase are specified in an odometric frame with its origin at the robot's initial position $\mathbf{P}_0$. In order to walk directly to the goal, some form of localization is required. We do not use an overhead camera to localize the robot or a prior map of the environment, but rather update odometry as the robot walks. The robot currently walks in an open-loop fashion, and while there is feedback on joint angles it is noisy and not at a high rate. While visual homing on the vehicle just found during the search phase is a viable option, we do not use it during this phase because of the practical limitation of the lack of a pannable neck. If the robot needs to go around an obstacle it will likely lose sight of vehicle due to the cameras' narrow field of view.

Our solution is visual odometry. Over the last decade visual odometry has become a common component of many indoor and outdoor robot navigation systems [29], [30], and it is particularly valuable on legged robots for which joint angle integration is error-prone [31], [32]. Visual odometry without scale ambiguity is typically accomplished using a stereo camera with feature triangulation, but because the

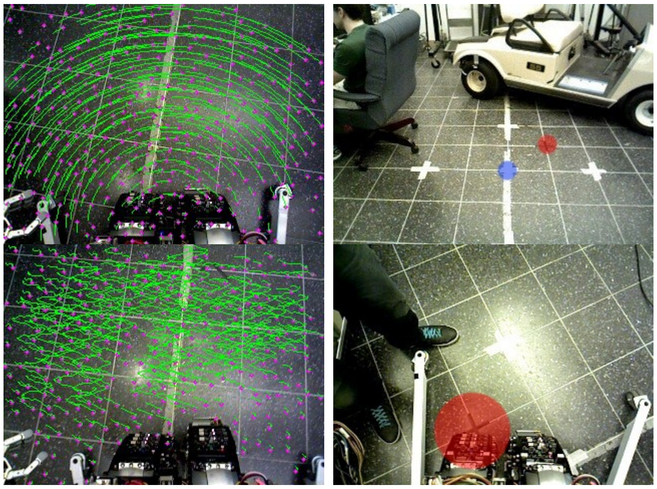
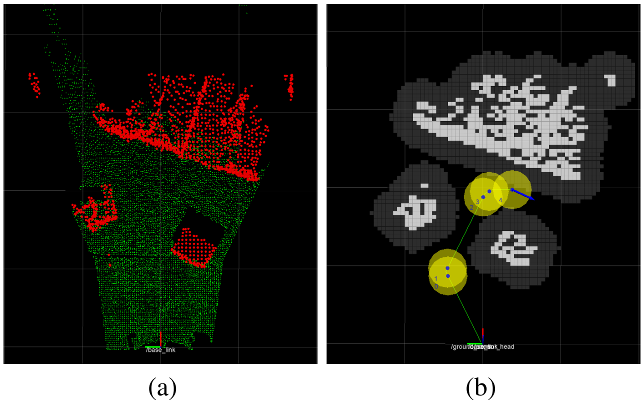


Fig. 6. Visual odometry for Hubo 2+ approach trajectory following. (Left) Feature tracks during waist panning and walking, clearly showing sway; (Right top) Robot was commanded to go to the spot indicated with the blue circle, then the spot with the red circle, and to turn to face parallel to the side of the vehicle. (Right bottom) The robot's final location.

depth cameras provide pixel-registered depth with RGB, this step is considerably simplified. We experimented with several existing libraries including [33], [34], but either because of efficiency or difficulty with proper parameter selection, wound up implementing our own approach.

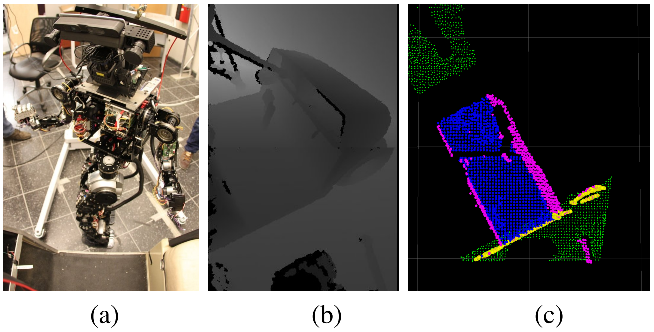
Our features are OpenCV Good Features to Track, and frame-to-frame match hypotheses are derived directly from Lucas-Kanade optical flow in pyramids. Outliers are filtered with robust fundamental matrix fitting, and tracks are extended for as many frames as possible. New features are added in every new frame where old tracks have ended. Motion is estimated between the set of tracked features at time t relative to their location in a *keyframe* up to several seconds before, where new keyframes are started when the number of viable tracks has dropped below threshold. The keyframe-relative motion estimates are computed from the 3-D locations of the features as inferred from the registered depth images, and it is these estimates which are integrated as the robot moves.

The Hubo robot walking motion is not currently controlled at the fine-grained level of footstep planning as in [19], but rather through modal directives such as "walk forward", "stop", "turn in place left", and so on. Under this regime the robot cannot turn while walking, and step lengths are a constant 5 cm each. To accommodate these limitations, we use a search-based planner built on straight and turn-in-place motion primitives [35], [36] which runs on a costmap generated from the obstacles found after ground plane extraction. This limits the curviness of generated paths greatly, and with some further filtering we obtain paths that avoid obstacles with relatively long straight segments. An example is shown in Fig. 7.



(a) (b)

Fig. 7. Screenshots of ROS `rviz` corresponding to scene shown in Fig. 3. (a) Ground/obstacle segmentation; (b) Obstacle costmap (white), inflated obstacles (gray), SBPL path (green) to entry point pose simplified to subgoals (yellow discs).



(a) (b) (c)

Fig. 8. (a) Hubo 2+ during docking phase of ingress on golf cart; (b) Combined depth image from robot's point of view; (c) Corresponding `rviz` screenshot showing segmented vehicle floor (blue) and edge (yellow) that guide docking motions.

V. Docking phase

At the end of the approach phase the robot is assumed to have stopped near the $\mathbf{P}_{entry}$ from Sec. III, with some error based on its initial pose and odometry. Whereas during approach we used an odometric coordinate system with its origin at the robot's initial pose, now it is convenient to switch to a vehicle-based coordinate system (recall that $+X$ is forward and $+Y$ is lateral toward the driver's side). Specifically, we set the origin to the forward end of the entry door. If the doorway can be visualized and tracked, we can directly measure the error between the actual robot pose and the entry pose as its position (x, y, θ) in door coordinates, where θ is the angle between the robot heading and the door line segment. The acceptable errors $(\Delta x, \Delta y, \Delta\theta)$ around the optimal entry pose determine an entry pose *target zone*. Experimentally and though simulation we have found that zone to be about 0.1 m $\times$ 0.1 m $\times 5°$.

If the robot is in the entry pose target zone at the end of the approach phase, we can skip directly to the *step phase*. Otherwise, the robot must reposition itself. We do this by directly tracking the doorway and performing position-based visual servoing on it to drive the error down. During this phase the robot is allowed to step forward and turn in place as during *approach*, but the step lengths are shortened (e.g. to 2.5 cm). The robot is further allowed to step backward and sidestep, also in the same shortened lengths. With the current walking controller these moves cannot be combined,

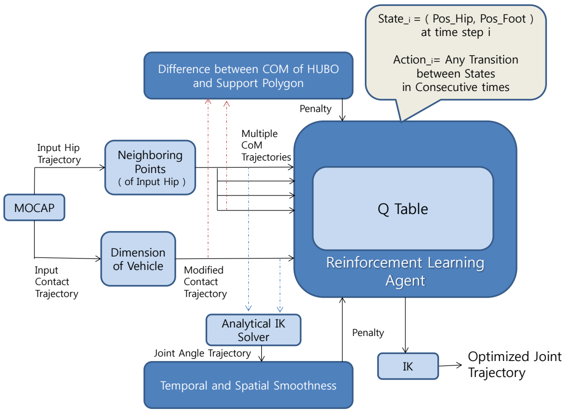


Fig. 9. Reinforcement learning agent-based algorithm for step trajectory generation

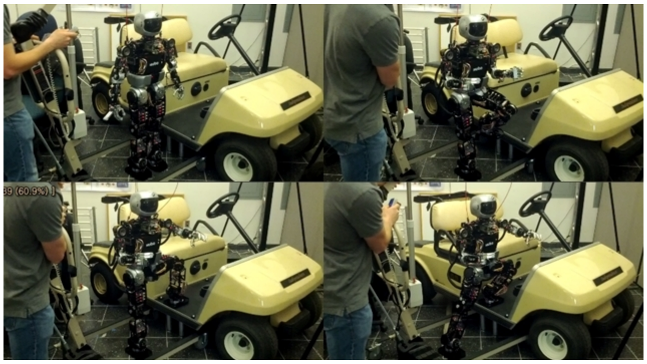


Fig. 10. Hubo 2+ demonstrating the validated trajectory for the initial step phase on the golf cart.

so the robot serially reduces first $\Delta\theta$, then Δx, then Δy.

For maneuvering so close to the vehicle, the robot "tucks" its arm close to its torso. Another issue is that head sway gives variable distance readings as the robot sidesteps. The head enters the target zone before the torso, so to prevent premature ending of the phase we require that visual odometry report no movement before allowing a final measurement. Also, there is some difficulty with the ground plane tracking locking onto the vehicle floor plane when there is no mechanism to reject height jumps during docking.

VI. Step Phase

To generate a step motion for the Hubo robot, we first produced an initial contact trajectory of end effectors which is a trace of foot movement starting from a designated ground position and arriving at a goal position in the vehicle. This trajectory was first based on motion capture of human hip and feet while executing the motion (the motion capture system is visible in Fig. 1).

The hip trajectory of captured human motion was assigned to a reinforcement learning algorithm-based optimization agent as an input center of mass (CoM) trajectory. Fig. 9 illustrates the trajectory optimization architecture. At each time step, the hip position of the input trajectory was allowed to change its values with in bounded ranges and this variance generated multiple states of each time step in learning algorithm. Thus, the goal of the learning agent is to choose an optimal sequence of states from these choices with respect to defined constraints. Further details are in [37].

Inputs of the optimization agent include only a CoM trajectory of hip and a contact trajectory of foot from the captured human motion. Since the hip trajectory allows variance of its values in learning agent and generated states of Q table in each time step, only foot trajectory has constant values at each time step. However, as we described above, position and rotation values of the foot trajectory could be modified and re-designed before optimization process starts. This allowance of the agent could adapt contacts of end effector according to dimension of the utility vehicle and could generate various output trajectories for a step motion. For example, by decreasing the maximum height of left and right foot in an input foot trajectory, the optimization agent could generate a step motion for Hubo with a smaller step height and still guarantee static balancing.

After testing the optimized trajectory in a simulation model for collision and quasi-static balancing, the validated trajectory was applied to the real Hubo 2+ robot on our test vehicle (a golf cart) using offline vehicle measurements as shown in Fig. 10. These are exactly the measurements provided by the perception module during the docking phase.

Hubo 2+'s hand strength is insufficient to consider gripping vehicle parts such as the dash or roof pillars to aid the step motion, but with the DRC-Hubo we have been able to make use of the latter. The full step phase for the DRC-Hubo in the golf cart is shown in Fig. 11. We have also been able to replicate this motion on a larger vehicle (a Polaris Ranger XP900) with the sensor head attached.

VII. Conclusion

We have a presented a set of modules related to the vehicle ingress task which run on the Hubo 2+ humanoid robot platform. Each of the modules works well and there is a loose integration between them, but there is much more work to be done to make the system work end-to-end seamlessly. In future work we plan to move to outdoor environments for testing and will thus be incorporating data from the color stereo cameras and tilt ladar on the DRC-Hubo sensor head.

Acknowledgments

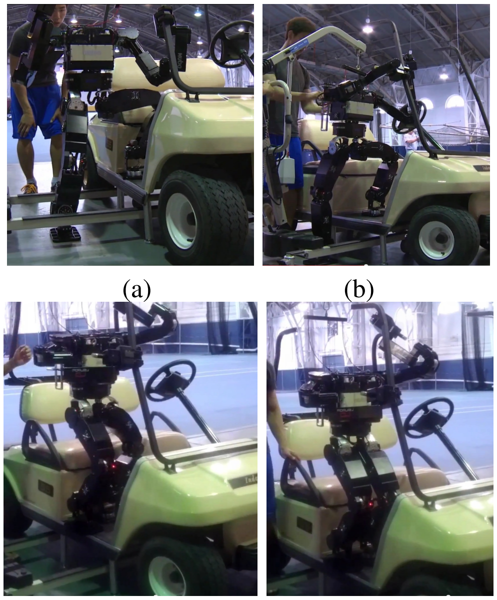
This work was supported in part by the Defense Advanced Research Projects Agency award #N65236-12-1-1005.

References

[1] DARPA, "DARPA Robotics Challenge website," 2012, available at http://darparoboticschallenge.org. Accessed December, 2012.

[2] M. Stark, M. Goesele, and B. Schiele, "Back to the Future: Learning Shape Models from 3D Cad Data," in *British Machine Vision Conference*, 2010.

[3] A. Karpathy, S. Miller, and L. Fei-Fei, "Object Discovery in 3D scenes via Shape Analysis," in *Proc. IEEE International Conference on Robotics and Automation*, 2013.



(a) (b)

(c) (d)

Fig. 11. DRC-Hubo demonstrating the full step phase on the golf cart. (a) Left foot step up with left hand grip; (b) Left foot shuffle, switch to right hand grip; (c) right foot up; (d) left hip rotate to parallel stance. The robot can sit down from this posture.

[4] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes," in *Asian Conference on Computer Vision*, 2012.

[5] J. Liebelt and C. Schmid, "Multi-View Object Class Detection With a 3D Geometric Model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[6] H. Koppula, A. Anand, T. Joachims, and A. Saxena, "Labeling 3D scenes for Personal Assistant Robots," in *Proc. RSS workshop on RGB-D Cameras*, 2011.

[7] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based Object Labeling in 3D Scenes," in *Proc. IEEE Inter. Conf. on Robotics & Automation*, 2012.

[8] Z. Marton, D. Pangergic, R. Rusu, A. Holzbach, and M. Beetz, "Hierarchical Object Geometric Categorization and Appearance Classication for Mobile Manipulation," in *Proc. IEEE-RAS Inter. Conf. on Humanoid Robots*, 2010.

[9] R. Rusu, Z. Marton, N. Blodow, A. Holzbach, and M. Beetz, "Model-based and Learned Semantic Object Labeling in 3D Point Cloud Maps of Kitchen Environments," in *Proc. IEEE/RSJ Inter. Conf. on Intelligent Robots & Systems*, 2009.

[10] R. Rusu, W. Meeussen, S. Chitta, and M. Beetz, "Laser-based Perception for Door and Handle Identification," in *Proc. Inter. Conf. on Advanced Robotics*, 2009.

[11] D. Hoiem, C. Rother, and J. Winn, "3d layoutcrf for multi-view object class recognition and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[12] H. Schneiderman and T. Kanade, "A statistical model for 3d object detection applied to faces and cars," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[13] D. Jang and M. Turk, "Car-rec: A real time car recognition system," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, 2011.

[14] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara, "Obstacle avoidance and path planning for humanoid robots using stereo vision," in *IEEE International Conference on Robotics and Automation*, 2004.

[15] T. Moulard, P. Alcantarilla, F. Lamiraux, O. Stasse, and F. Dellaert, "Reliable indoor navigation on humanoid robots using vision-based localization," in *IEEE International Conference on Robotics and Automation*, 2013.

[16] S. Oßwald, J. Gutmann, A. Hornung, and M. Bennewitz, "From 3D Point Clouds to Climbing Stairs: A Comparison of Plane Segmentation Approaches for Humanoids," in *Proc. of the IEEE-RAS Inter. Conf. on Humanoid Robots*, 2011.

[17] K. Okada, S. Kagami, M. Inaba, and H. Inoue, "Plane segment finder: Algorithm, implementation and applications," in *Proc. IEEE Inter. Conf. on Robotics & Automation*, 2001.

[18] J. Gutmann, M. Fukuchi, and M. Fujita, "Stair climbing for humanoid robots using stereo vision," in *Proc. IEEE/RSJ Inter. Conf. on Intelligent Robots & Systems*, 2004.

[19] J. Chestnutt, Y. Takaoka, K. Suga, K. Nishiwaki, J. Kuffner, and S. Kagami, "Biped navigation in rough environments using on-board sensing," in *Proc. IEEE/RSJ Inter. Conf. on Intelligent Robots & Systems*, 2009.

[20] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *Inter. Journal of Robotics Research*, vol. 31, no. 11, pp. 1251–1262, 2012.

[21] V. Sanchez and A. Zakhor, "Planar 3D Modeling of Building Interiors from Point Cloud Data," in *ICIP*, 2012.

[22] K. Bouyarmane, J. Vaillant, F. Keith, and A. Kheddar, "Exploring Humanoid Robots Locomotion Capabilities in Virtual Disaster Response Scenarios," in *Proc. IEEE International Conference on Humanoid Robots*, 2012.

[23] Z. Qiu, A. Escande, A. Micaelli, and T. Robert, "A Hierarchical Framework for Realizing Dynamically-stable Motions of Humanoid Robot in Obstacle-cluttered Environments," in *Proc. IEEE International Conference on Humanoid Robots*, 2012.

[24] 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.

[25] 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.

[26] R. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Inter. Conf. on Robotics & Automation*, 2011.

[27] C. Rasmussen, K. Yuvraj, R. Vallett, K. Sohn, and P. Oh, "Towards Functional Labeling of Utility Vehicle Point Clouds for Humanoid Driving," in *IEEE Int. Conf. on Technologies for Practical Robot Applications*, 2013.

[28] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," in *ACM Symposium on User Interface Software & Technology*, 2011.

[29] K. Konolige, M. Agrawal, and J. Sola, "Lare Scale Visual Odometry for Rough Terrain," in *International Symposium on Robotics Research*, 2007.

[30] N. Seegmiller and D. Wettergreen, "Optical Flow Odometry with Robustness to Self-shadowing," in *IEEE International Conference on Intelligent Robots and Systems*, 2011.

[31] J. Ma, S. Susca, M. Bajracharya, L. Matthies, M. Malchano, and D. Wooden, "Robust Multi-Sensor Day/Night 6-DOF Pose Estimation for a Dynamic Legged Vehicle in GPS-Denied Environments," in *Proc. IEEE International Conference on Robotics and Automation*, 2012.

[32] A. Chilian, H. Hirschmuller, and M. Gorner, "Multisensor Data Fusion for Robust Pose Estimation of a Six-Legged Walking Robot," in *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2011.

[33] A. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera," in *International Symposium on Robotics Research*, 2011.

[34] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2013.

[35] M. Likhachev. (2013) Ros search-base planning library stack. [Online]. Available: http://www.ros.org/wiki/sbpl

[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] K. Sohn and P. Oh, "Applying human motion capture to design energy-efficient trajectories for miniature humanoids," in *Proc. IEEE/RSJ Inter. Conf. on Intelligent Robots & Systems*, 2012.