2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Automated Perception of Safe Docking Locations with Alignment Information for Assistive Wheelchairs

Siddarth Jain[1] and Brenna Argall[2]

Abstract— **There are basic manuvering tasks with a powered wheelchair, like docking under a table and passage through a doorway or narrow hallway, which can be difficult for users with severe motor impairments—not only because of limitations in their own motor control, but also because of limitations in the control interfaces available to them. Robot automation can help transfer some of this control burden from the user to the machine. This work presents an algorithm for the automated detection of safe docking locations at rectangular and circular docking structures (tables, desks) with proper alignment information using 3D point cloud data. The safe docking locations can then be provided as goals to an autonomous path planner, within the context of providing adaptive driving assistance for powered wheelchair users. We evaluate the performance of our algorithm with systematic testing on several docking structures, observed from varied viewpoints.**

I. INTRODUCTION

Powered wheelchairs provide a mobility solution for people unable to operate a manual wheelchair, for reasons of strength or impairment. However, operation of a powered wheelchair can still be a difficult, tedious or challenging task. The operator must be able to accurately sense their environment and, if using traditional control interfaces, translate their mobility desires into continuous joystick commands for the wheelchair. Those who lack the requisite motor skills, strength, or visual acuity are often left unable to operate a powered wheelchair. In a survey of clinicians [1] 10% of patients who receive powered wheelchair training find it extremely difficult or impossible to use, and the number rises to 40% if they are asked about the maneuvering tasks associated with completing Activities of Daily Living (ADL).

In order to provide assistance to this population, several "smart" wheelchairs have been proposed, that leverage robotics automation to assist in driving a powered wheelchair. Some features of a smart wheelchair might be to sense the environment, estimate the intention of a user or execute control strategies that achieve complex operations through a smooth and safe set of movements. Smart wheelchairs often target providing assistance for ADL behaviors that pose challenges for steering and manuvering in constrained, narrow or cluttered spaces—that require increased dexterity to avoid collisions. Two spatially-constrained assistance behaviors that receive much attention within the literature are doorway-passage and docking. While multiple works address the autonomous perception required for doorway-passage [2]–[4], fully autonomous perception for docking assistance is minimally addressed and remains an open research question.

To identify suitable docking surfaces in the environment and then assess them for safety is a challenging perception task, and docking furthermore requires accurate alignment or orientation information. One possible solution is a machine learning system trained on various docking surface configurations—however, re-training is likely required if conditions like furniture locations or scene clutter change. Furthermore, when image data is used, it is affected by changing lighting conditions and contrast. By comparison, our approach to identify docking surfaces is based on geometric information, computed from depth data, which makes it invariant to scene or lighting changes.

We propose two approaches for determining docking locations, which we integrate within a single algorithm to get a list of safe locations available in the input scene. The first approach involves a sequential search around the edges of all the candidate structures (tables/desks) in the scene (Fig. 1, top-right). The second approach is formulated based on a restaurant setting or a dinner table scene in which objects such as plates, bowls and mugs are on a table (Fig. 1, bottom-right). Our insight is that such objects frequently mark desirable docking locations. In both cases, we then scan for free space underneath the candidate locations, which results in a list of potential safe docking locations.

The larger goal of this work is to develop a complete docking assistance system for our smart wheelchair (Fig. 1, left), which enables the robot to automatically perceive safe docking locations in the environment, with proper pose alignment. Most indoor docking structures are either rectangular or circular in geometry, and so our approach handles the perception of both—at a vertical height such that a person seated in a powered wheelchair (which has fairly standardized dimensions) would be able to interact with the surface. Furthermore, our algorithm is operable in novel environments, does not require any fiducial markers or alteration of the environment, and is designed to work easily on any powered wheelchair or mobile robot platform.

In the next section, we present background information and review related work. Section III details our algorithm. Section IV describes our validation approach and presents the experimental results. In the final section we conclude with directions for future work.

[1]Siddarth Jain is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA siddarthjain2017@u.northwestern.edu

[2]Brenna Argall is jointly appointed with the Faculty of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA and the Rehabilitation Institute of Chicago, Chicago, IL 60611, USA brenna.argall@northwestern.edu

978-1-4799-6934-0/14/$31.00 ©2014 IEEE

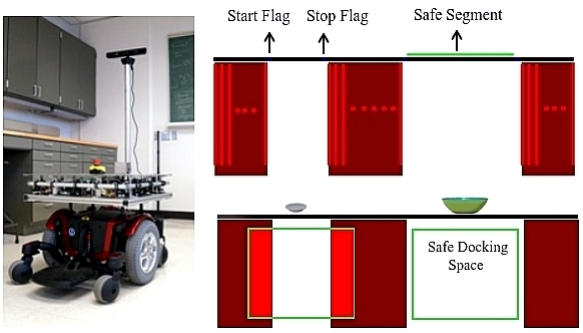


Fig. 1. *Left*: Illustration of our smart wheelchair with Kinect. *Right*: The sequential search approach (top) and object anchoring approach for a restaurant/dinner table setting (bottom).

II. RELATED WORK

Here we discuss related work on perception for docking in the context of smart wheelchairs and mobile robots. Many smart wheelchair works recognize docking as a desired behavior or mention it as a goal for future work [5]–[13].

Approaches that address robot docking often make use of fiducial markers. Weber *et al.* [14] present a solution for robot docking based on neural networks and computer vision, that used an orange fruit placed on the edge of a table as a visual fiducial. Martínez-Marín & Duckett [15] mark an orange stripe on the edge of a table, which they identify by color segmentation in HSV space. A landmark-based docking approach using an omnidirectional camera is presented by Muse *et al.* [16], in which image processing is used to locate a landmark placed beneath the table.

Other approaches simplify the problem by focusing on customized docking structures. Bailey *et al.* [17] use a laser scanner and geometric constraints to find the corners of a rectangular container of known dimensions, which also is used to calculate the relative container pose. Ren *et al.* [7] target docking a wheelchair at a U-shaped bed. For bed recognition, they place a T-shaped fiducial on the bed (similar to [14], [15]). Instead of docking at furniture, Sermeno-Villalta & Spletzer [8] provide an approach for docking a wheelchair onto a vehicle lift platform, using visual feedback control and fiducials for image processing and tracking.

A handful of works do not rely on environment modifications like fiducials or customization. Nuttin *et al.* [9] estimate user intention based on joystick signals and detect tabletops to provide assistance for wheelchair docking. To address the problem of an identified table with an unclear docking location, their algorithm selects a location somewhere in between the edges as a compromise. There also are assistive wheelchair works that place the identification of docking locations entirely with the human [10]–[12].

There are approaches that use 3D data to find supporting planes in the scene; an approach similar to our work. The application of a Mixture of Gaussians probabilistic model built using Expectation-Maximization (EM) is suggested by Williamson *et al.* [18] to find possible docking objects by segmenting data from a LIDAR 3D laser range-finder for modeling plane-like data clusters—which however requires knowing *a priori* the number and types of clusters and is subject to finding poor local minimum solutions. Wei *et al.* [13] present a 3D semantic map based approach using RGB-D data and the Random Sample Consensus (RANSAC) algorithm to extract planes in the scene, which are matched to entries in a model library to identify objects. The work suggests using feature extraction to identify free docking positions and alignment information, however the implementation details are missing.

We are interested in a perception system that provides alignment information, so that the robot approaches the docking surface perpendicularly. There are approaches which focus on estimating alignment using optical flow [19]. Our approach (Sec. III) uses the geometric structures of the identified objects, similar to [17]—however without the requirement of knowing the structure dimensions *a priori*.

Most of the approaches mentioned above provide a limited solution to the docking problem by making use of fiducial markers or landmarks ([14]–[16]), or have simplified the problem by focusing on customized docking structures ([7], [17]). Some provide a general solution to identify the docking structures in the scene ([9], [13], [18]), but the search and determination of precise safe locations and orientation information is unclear in these works. Furthermore, approaches based on image data are susceptible to problems of changing contrast and vision distortions, which have been identified as bottlenecks in some systems ([14]). We take the opportunity here to restate that our work does address docking perception as well as the safe position and alignment determination, and is also free from the usage of landmarks or fiducial markers.

III. ALGORITHM

We now present our algorithm for the automated perception of safe docking locations. Our algorithm first identifies candidate docking surfaces (Sec. III-A), classifies them as rectangular or circular (Sec. III-B) and extracts their edges/boundary (Sec. III-C, D), and then scans along it for candidate docking locations which furthermore are assessed for safety (Sec. III-E).

The input to our algorithm is a point cloud of the scene, where each point is represented by a tuple containing its 3D position in camera coordinates $\langle x, y, z \rangle$. The Kinect is mounted on the wheelchair 130*cm* above the ground (Fig. 1, left). In the camera coordinate frame the Y-axis is perpendicular to the ground, the Z-axis is back-to-front, and the X-axis is left-to-right. This information is then transformed onto the world coordinate frame and stored in the list χ_l of safe docking locations.

A. Candidate Docking Structure Extraction

The first challenge is to identify all of the candidate docking structures, such as tables and desks, present in the scene. To identify such candidates, we use RANSAC to search for all the planar patches that satisfy the following conditions: (i) they are perpendicular to the Y-axis, $i.e$ the vertical axis of the camera frame; (ii) their height is at least

(a) (b) (c)

Fig. 2. Example of the candidate structure shape classification by circle detection on the 2D-grid representation of the convex hull. Point Cloud of a scene with a round table *(a)* and a rectangular table *(b)*. Convex hull marked in green points. *(c)* Circle estimated from the circular table convex hull.

τ above the ground plane; and (iii) their point cloud size is larger than η. In our implementation, we used τ = 28*in* and η = 3000. Each such planar patch is then clustered to form a set of planar regions P_c.

Once we have the candidate set P_c, the goal is to search each region $P_c^i \in P_c$ for the set of locations χ_l that are safe for docking, where $\chi_l^i \in \chi_l$ is a tuple containing the 3D position $\langle x_l^i, y_l^i, z_l^i \rangle$ of a safe docking location.

B. Candidate Structure Shape Classification

We restrict our search for candidate surface structures to shapes that are either rectangular or circular in geometry. We first reason about the shape (rectangular or circular) of the candidate structures in P_c, since the approach we follow to identify safe docking locations differs between the two.

To differentiate between the two shapes, we base our approach on the application of the Hough Transform technique to detect circle models. Instead of directly using the candidate structure point cloud data to run the Hough Transform, we extract the convex hull H_c^i for each $P_c^i \in P_c$ and create a 2D-grid representation $G_{x,z}^i$ of P_c^i—using each pair $\langle x_c^{i,j}, z_c^{i,j} \rangle \in H_c^i$ such that the grid size equals the area of the region P_c^i, and each grid cell that corresponds to a pair $\langle x_c^{i,j}, z_c^{i,j} \rangle$ is set to 1 with all others being set to 0. Note that quantization is performed in order to index the grid cells, and the y-coordinates are not considered here as we are interested in identifying candidate structures that are planar in geometry and perpendicular to the Y-axis. The 2D-grid representation $G_{x,z}^i$ simplifies running the Hough Transform and also reduces the time complexity. Based on the detection result of the Hough Transform we classify each $P_c^i \in P_c$ to create a set of rectangular candidates R_c and a set of circular candidates C_c. An illustration of the circle detection on a round table structure is show in Figure 2.

C. Corner and Edge Extraction for Rectangular Structures

We detect each corner in $R_c^i \in R_c$ for two reasons: (i) to use as an anchor to initiate our search for safe docking locations, and (ii) to exclude corners from the set of locations χ_l (since it is unlikely that a user will want to dock at a corner location). We implemented and compared two approaches to identify corners.

The first approach is based on finding a rotated rectangle of the minimum area enclosing the input point set in R_c^i. Once we have the the minimum-area bounding rectangle for R_c^i we compute the four corners using basic trigonometry.

TABLE I

PERFORMANCE COMPARISON FOR CORNER DETECTION

Rectangle Fit Approach			
Offset	Avg.Time (*ms*)	True Corners	Spurious corners
0 degrees	0.20	40	0
45 degrees	0.21	40	0
-45 degrees	0.19	36	4

Hough Transform based Approach			
Offset	Avg.Time (*ms*)	True Corners	Spurious corners
0 degrees	17.07	36	3
45 degrees	18.23	32	3
-45 degrees	16.08	37	3

The second approach is based on the application of the Hough Transform to model lines in candidate region R_c^i and computing the line intersections to find the corner locations. We again create a 2D-grid representation $G_{x,z}^i$ as was done in Section III-B, but this time we use the point cloud of R_c^i instead of using the convex hull. Furthermore, we apply morphological operations using a 3x3 square structuring element, $Mat(3,3)$, on $G_{x,z}^i$ to extract a smooth boundary for the candidate region R_c^i. Specifically, dilation $G_{x,z}^i \oplus Mat(3,3)$ is performed to blend the sparse regions in the point cloud data, followed by an erosion operation $G_{x,z}^i \ominus Mat(3,3)$. Boundary extraction is done by subtracting the result of the erosion operation from the dilation result, which gives the desired smooth boundary representation $B_{x,z}^i$ of the candidate structure. $B_{x,z}^i$ is then used to model lines by application of the Hough Transform.

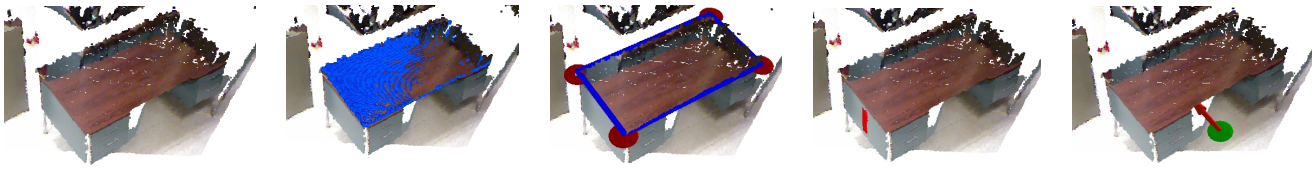
A performance analysis of the two approaches for corner detection found the rectangle-fit approach to be superior to the Hough Transform approach (Tbl. I, 10 runs for each of 3 viewing angles: 0°, 45° and -45° offset, so that the camera faces the edge and the two corners). Based on this evaluation, the rectangle-fit approach was integrated into our algorithm.

Once the four corners $\langle \kappa_c^{i,1}, \kappa_c^{i,2}, \kappa_c^{i,3}, \kappa_c^{i,4} \rangle$ for each R_c^i have been identified (Fig. 3c), line models $\langle L_c^{i,1}, L_c^{i,2}, L_c^{i,3}, L_c^{i,4} \rangle$ are computed between them. The edge points are computed according to (1), by progressively varying parameter λ. In this way the algorithm is able to identify the edges of the rectangular candidate structures present in the scene.

$$L_c^{i,j,k} = \kappa_c^{i,j} + \lambda^k \cdot (\kappa_c^{i,j} - \kappa_c^{i,l}) \tag{1}$$

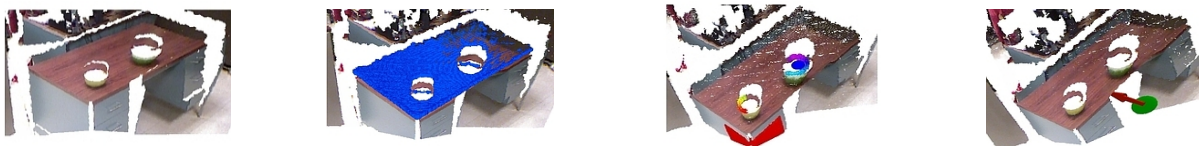
D. Boundary Extraction for Circular Structures

In order to check the circular candidate structures C_c for the clearance needed for safe docking, we first extract the boundary of each $C_c^i \in C_c$. We use RANSAC on the convex hull representation of C_c^i to find the model parameters of the circle. Having the center a and radius r for each C_c^i, the points $\langle Cb_c^{i,x,k}, Cb_c^{i,z,k} \rangle$ on the boundary are computed according to (2) and (3), by progressively varying parameter θ. The y-coordinate of each point, $Cb_c^{i,y,k}$, is then recovered from the point cloud data of C_c^i. In this way the algorithm is able to extract the boundary of circular candidate structures



(a) (b) (c) (d) (e)

Fig. 3. Example of the sequential search approach. *(a)* Point Cloud of the scene with image overlay. *(b)* Candidate surface extraction (blue region). *(c)* Corner locations (red markers) and extracted edges represented by line models (dark blue lines). *(d)* 3D box stripe (vertical red stripe) running along edges searching for free segments. *(e)* Identified docking location (green marker) and docking pose (red arrow).



(a) (b) (c) (d)

Fig. 4. Example of the object anchoring approach. *(a)* Point Cloud of the scene with image overlay. *(b)* Candidate surface extraction (blue region). *(c)* Object Clusters (rainbow-colored circles), Hough Circle Detection (blue marker) and 3D box (red box). *(d)* Identified docking location (green marker) and docking pose (red arrow).

present in the scene.

$$Cb_c^{i,x,k} = a_c^{i,x} + r_c^{i,x} \cdot \sin(\theta^k) \quad (2)$$

$$Cb_c^{i,z,k} = a_c^{i,z} + r_c^{i,z} \cdot \cos(\theta^k) \quad (3)$$

E. Identification of Safe Locations

The next step is to scan for the safe docking locations for each of the candidate $P_c^i \epsilon P_c$. We propose two approaches that both make use of an oriented 3D box that we place or slide underneath the candidate structure, depending on the candidate edges or object clusters. We count the number of cloud points falling inside this 3D box and base our safety decision on this information. The size of the box is set based on table specifications in the Americans with Disabilities Act (ADA)[1] (26×30×19*in* h×w×d).

1) Sequential Search—Identifying Safe Segments: For each candidate in P_c, we scan either along the four edges $\langle L_c^{i,1}, L_c^{i,2}, L_c^{i,3}, L_c^{i,4} \rangle$ or the boundary Cb_c^i. The 3D box is placed under the edge locations to find free segments $S_c^{i,j}$ that would be safe for the width of the wheelchair. Here, the 3D box slides along the four edges or the boundary of the structure candidate and is checked iteratively in sequential vertical strips (Fig. 3d). A safe docking location χ_c^i is marked as the midpoint of the box.

For alignment information for rectangular structures, we compute the docking pose orientation as the perpendicular direction to the corresponding edge that constitutes the safe segment (Fig. 3e). For circular structures, we calculate the direction of the vector pointing from the docking location to the center of the circle.

2) Restaurant/Dinner Table Setting—Object Anchoring: For each structure candidate in P_c (Sec. III-A), we find object clusters O_c^i present on the candidate surface. To extract object clusters, we select the cloud points within a predefined threshold ($2cm \leq \Gamma \leq 15cm$) above the surface and identify the points that fall within the convex hull of the candidate surface. A similar approach is taken by Rusu *et al.* [2] to extract door handles. Next, object clusters are extracted by performing Euclidean clustering. We then base our candidate shape search on the identification of circle models in the object clusters O_c^i (Fig. 4c)—since the objects considered are bowls, plates and mugs, we assume that the object clusters will have a circular shape. A 2D-grid representation $G_{x,z}^i$ for each of the object cluster O_c^i is created as described in Section III-B, which is then used to detect a circle by the application of the Hough Transform. If a circle is detected, we place a 3D box underneath that cluster center to search for a safe docking space (Fig. 4c). For alignment information we identify the closest point on the edge or boundary (depending on the candidate structure shape) to the object of interest and use the method described in Section III-E.1 to compute the orientation for docking alignment (Fig. 4d).

[1]Americans with Disabilities Act: `http://www.ada.gov/2010ADAstandards_index.htm`

IV. EXPERIMENTAL RESULTS

A systematic approach to collecting data on docking structures was used to characterize the algorithm performance. The algorithm was validated for different configurations on four types of docking structures: a rectangular table, desk, workbench, and a round table (Fig. 5). Experiments were carried out on an Intel Core i7 Quad-Core processor PC with 8GB of RAM and running Ubuntu 12.04.

(a) (b) (c) (d)

Fig. 5. Docking structures used for evaluation. *(a)* Table *(b)* Desk *(c)* Workbench and *(d)* Round table.

TABLE II

PERFORMANCE EVALUATION AT VARYING OFFSET ANGLES ON DIFFERENT DOCKING STRUCTURES AND CONFIGURATIONS

Structure & Configuration	**Offset Angle (°)**	**# Safe locations**	**True Positives**	**True Negatives**	**False Positives**	**False Negatives**
Table - 4 chairs	0 / 45 / -45	40 / 40 / 40	37 / 32 / 36	20 / 30 / 10	0 / 5 / 2	3 / 8 / 4
Desk	0 / 45 / -45	10 / 10 / 10	10 / 10 / 9	- / - / -	0 / 0 / 1	0 / 0 / 1
Desk - 2 objects	0 / 45 / -45	10 / 10 / 10	9 / 8 / 8	10 / 10 / 10	0 / 0 / 0	1 / 2 / 2
Workbench	0 / 45 / -45	20 / 20 / 20	19 / 17 / 18	- / - / -	0 / 2 / 0	1 / 3 / 2
Workbench - 2 objects	0 / 45 / -45	20 / 20 / 20	17 / 17 / 16	- / - / -	0 / 0 / 0	3 / 3 / 4
Workbench - 1 object, 1 chair	0 / 45 / -45	10 / 10 / 10	9 / 8 / 8	10 / 10 / 10	0 / 0 / 0	1 / 2 / 2
Circular table - 3 chairs	-	10	9	20	0	1
Circular table - 2 objects,3 chairs	-	10	8	10	0	2

To evaluate shape classification (Sec. III-B) we tested each of the four docking structures for 10 runs of the algorithm. It was observed that they were correctly classified by the approach. Table III summarizes the results.

To test the sequential search and the object anchoring approach, different configurations that might be commonly encountered were evaluated. For rectangular structures each configuration was tested from three different views: 0°, 45° and -45° offset. Each combination of docking configuration and viewing angle was evaluated over 10 cycles. The point cloud stream was paused at each cycle of the algorithm to facilitate the online manual counting of correctly identified locations, falsely identified locations and missed locations.

The point cloud data underneath the candidate docking structures often is visible only for the sides closest to the sensor. The algorithm considers locations to be safe until it has concrete data to say otherwise. Accordingly, docking locations identified on far sides of the tables are occasionally marked as safe when in fact they are not free. We consider this risk to be minimal however, since the wheelchair necessarily needs to drive near a location to dock at it, at which point the necessary sensor data becomes available. Such cases were not counted as false positives in the evaluation.

In each of the described cases, the algorithm performed quite well and succeeded in detecting the safe docking locations, with relatively few missed locations. Table II presents the performance data for each of the testing configurations, and the angles at which they were evaluated. Figure 6 presents a number of examples of the algorithm at work on variety of docking structure configurations.

In order to achieve good run-time speed, the RANSAC algorithm was used for plane-fitting, which occasionally lead to somewhat strange surface selections and resulted in false positive identifications of docking locations—particularly for the sequential search approach, as it depends on the corner detection accuracy.

TABLE III

PERFORMANCE EVALUATION FOR SHAPE CLASSIFICATION

Docking Structure Type	Number	Classified as	
		Rectangular	Circular
Table	10	10	0
Desk	10	10	0
Workbench	10	10	0
Round table	10	0	10

The effective working distance range of the algorithm was observed to be approximately $3m$. At distances greater than this, the horizontal planar segmentation became less reliable since the point cloud data returned by the Kinect decreased in accuracy. However, this operational range is useful and appropriate for the intended task of assisting wheelchair users, as the docking assistance is mostly required when the user is in close proximity to the docking structure.

A series of videos were recorded demonstrating the performance of the algorithm.[2] Finally, we note that a quantitative comparison to other docking approaches is not provided, since fully autonomous perception for docking assistance is not addressed by any other works, as they: make use of landmarks, fiducial markers or customized docking structures; do not check for safety; or do not provide the necessary alignment information.

V. CONCLUSIONS AND FUTURE WORK

We have introduced in this paper a novel method for the autonomous detection of safe docking locations using 3D point cloud data, and without any visual fiducial or environment customization requirements. Through evaluations on differently shaped docking structures in varying configurations, our algorithm has demonstrated good performance and was shown to be effective in the identification of safe and oriented docking locations. Finding safe docking locations is important in the context of assistive wheelchairs, and also for autonomous mobile robots. By detecting safe docking locations along with alignment information, custom trajectories can be planned and executed by a path planner to achieve the docking maneuver. Our hope is to benefit smart wheelchair technology by realizing the difficult problem of docking assistance. Future work includes dealing more robustly with the plane-fitting portion of the algorithm by considering different sample consensus methods, as it accounts for many of the reported errors in perception. We would also like to explore ways to measure the confidence in the perceived docking locations and to infer the user's preferred location using only the interface used to drive the wheelchair.

[2]The videos can be found at `http://smpp.northwestern.edu/research/argallab/video.html`

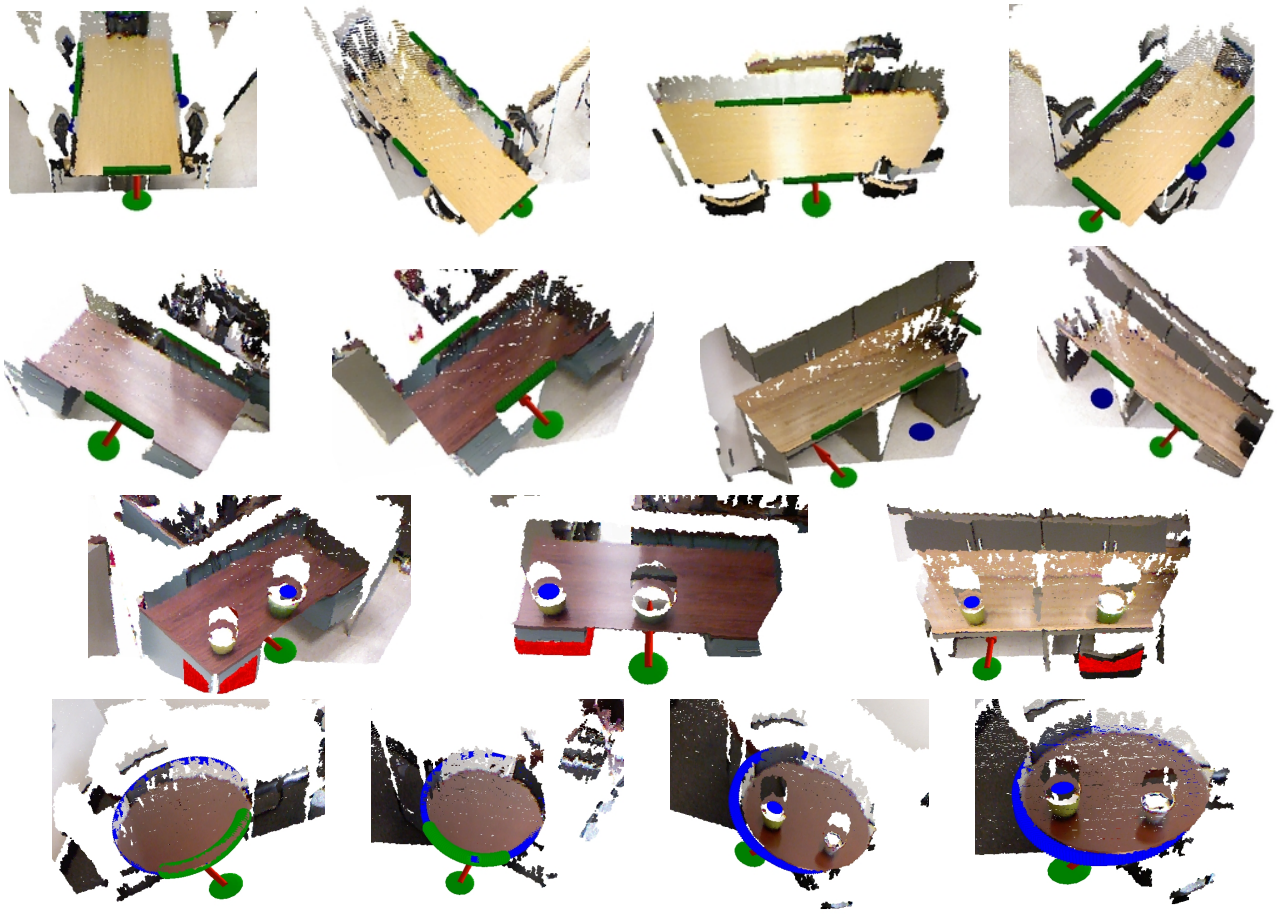


Fig. 6. Examples of the algorithm working correctly on a variety of docking structure configurations tested from varied viewpoints. The green marker indicates the preferred docking location and the dark blue marker indicates all other docking locations, the red arrow marks the pose orientation (for alignment information) and blocked space is marked by a red 3D box.

REFERENCES

[1] L. Fehr, W. Langbein, and S. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey.," *Development*, vol. 37, no. 3, pp. 353–360, 2000.

[2] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz, "Laser-based perception for door and handle identification," in *Proceedings of ICAR*, 2009.

[3] F. Cheein, C. De La Cruz, T. Bastos, and R. Carelli, "Slam-based cross-a-door solution approach for a robotic wheelchair," *International Journal of Advanced Robotic Systems*, vol. 7, no. 2, pp. 155–164, 2010.

[4] M. Derry and B. Argall, "Automated doorway detection for assistive shared-control wheelchairs," in *Proceedings of ICRA*, 2013.

[5] F. Leishman, O. Horn, and G. Bourhis, "Multimodal laser-vision approach for the deictic control of a smart wheelchair," in *Ambient Assistive Health and Wellness Management in the Heart of the City*, pp. 98–107, Springer, 2009.

[6] D. Vanhooydonck, E. Demeester, A. Hüntemann, J. Philips, G. Vanacker, H. Van Brussel, and M. Nuttin, "Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model," *Robotics and Autonomous Systems*, vol. 58, no. 8, pp. 963–977, 2010.

[7] Y. Ren, W. Zou, H. Fan, A. Ye, K. Yuan, and Y. Ma, "A docking control method in narrow space for intelligent wheelchair," in *Proceedings of ICMA*, 2012.

[8] H. Sermeno-Villalta and J. Spletzer, "Vision-based control of a smart wheelchair for the automated transport and retrieval system (ATRS)," in *Proceedings of ICRA*, 2006.

[9] M. Nuttin, D. Vanhooydonck, E. Demeester, and H. Van Brussel, "Selection of suitable human-robot interaction techniques for intelligent wheelchairs," in *Proceedings of ROMAN*, 2002.

[10] R. C. Simpson, D. Poirot, and F. Baxter, "The Hephaestus smart wheelchair system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 118–122, 2002.

[11] G. Wasson, P. Sheth, C. Huang, and M. Alwan, "Intelligent mobility aids for the elderly," in *Eldercare Technology for Clinical Practitioners*, pp. 53–76, Springer, 2008.

[12] T. Carlson and J. d. R. Millán, "Brain-controlled wheelchairs: a robotic architecture," *IEEE Robotics and Automation Magazine*, vol. 20, no. EPFL-ARTICLE-181698, pp. 65–73, 2013.

[13] Z. Wei, W. Chen, and J. Wang, "3D semantic map-based shared control for smart wheelchair," in *Intelligent Robotics and Applications*, pp. 41–51, Springer, 2012.

[14] C. Weber, S. Wermter, and A. Zochios, "Robot docking with neural vision and reinforcement," *Knowledge-Based Systems*, vol. 17, no. 2, pp. 165–172, 2004.

[15] T. Martínez-Marín and T. Duckett, "Fast reinforcement learning for vision-guided mobile robots," in *Proceedings of ICRA*, 2005.

[16] D. Muse, C. Weber, and S. Wermter, "Robot docking based on omnidirectional vision and reinforcement learning," *Knowledge-Based Systems*, vol. 19, no. 5, pp. 324–332, 2006.

[17] T. Bailey, E. Nebot, J. Rosenblatt, and H. Durrant-Whyte, "Behaviour-based docking using the DAMN arbiter," in *Proceedings of ACRA*, 1999.

[18] M. Williamson, R. Murray-Smith, and V. Hansen, "Robot docking using Mixtures of Gaussians," in *Advances in Neural Information Processing Systems 11*, 1999.

[19] N. Barnes and G. Sandini, "Direction control for an active docking behaviour based on the rotational component of log-polar optic flow," in *Proceedings of ECCV*, 2000.